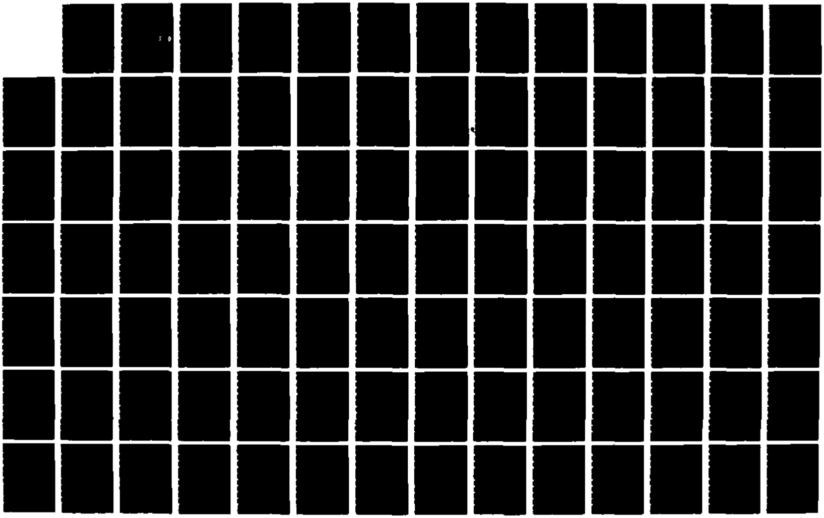
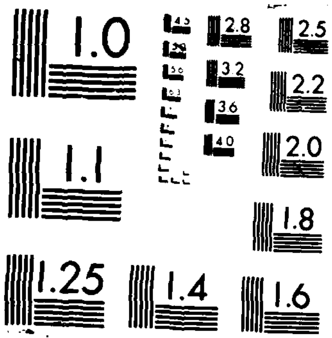
AD-A185 869 THE CLASSIFICATION PROBLEM OF FINITE RINGS BY COMPUTABLE MEANS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH W A KIELE 1987 1/2

UNCLASSIFIED AFIT/CI/NR-87-118T F/G 12/1 NL





ROCOPY RESOLUTION TEST CHART

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A185 869

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 87-118T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Classification Problem of Finite Rings by Computable Means		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William Albert Kiele		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: North Carolina State University		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433-6583		12. REPORT DATE 1987
		13. NUMBER OF PAGES 131
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

LYNN E. WOLAVER 25 Oct 87
Dean for Research and Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

87 10 28 180

118

The Classification Problem of Finite Rings

by Computable Means

by

William Albert Kiele

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Department of Mathematics

Raleigh

1987

Approved By:

Ernest Stitzinger

Kwangil Koh

Chairman of Advisory Committee

87 8 0

Abstract

KIELE, WILLIAM ALBERT, Maj, USAF. "The Classification Problem of Finite Rings by Computable Means" (Under the Direction of Kwangil Koh). 1987, Ph.D., North Carolina State University.

The purpose of this ~~paper~~ is to establish a constructive method for testing when two given finite rings are isomorphic. Currently published theory has classified a significant number of finite rings; however, "idealized" representatives are almost always used, with no provision for determining which isomorphism class an arbitrary ring belongs. The new results are as follows:

1. Two rings are isomorphic if and only if a specific system of quadratic equations is satisfied. This system, and a method of attacking it, were developed by the author.
2. As a corollary to the preceding result, there exists a system of linear equations that positively identify whether or not a ring R possesses a 1. The system also shows how to change a ring's basis so that 1 becomes a basis element. Some tests for existence of other idempotents besides 1 are shown.
3. Some old and new results in classifying finite rings of small rank are obtained with the help of theory developed in this paper.

REFERENCES

[Ba]: Baumgartner, K. "Bemerkungen Zum Isomorphieproblem der Ringe", *Monatsch. Math.* 70 (1966), pp.299-308.

[Bo]: Turbo PASCAL™ is a trademark of Borland Int'l.

[BP]: Beker, Henry and Fred Piper, Cipher Systems, John Wiley & Sons, 1982.

[Ca]: Carmichael, A.D. Introduction to the Theory of Groups of Finite Order, Ginn and Co., 1937.

[Fu]: Fuchs, L. Infinite Abelian Groups, vol II, Academic Press, 1973.

[Ja]: Jacobsen, N. The Structure of Rings, American Math. Soc. Colloquium Publications, vol 37, 1964.

[KP]: Kruse, R.L. and David T. Price, Nilpotent Rings, Gordon and Breach Science Publishers, 1969.

[MM]: Meyer, Carl H. and Stephen M. Mattyas, Cryptography: A New Dimension in Computer Data Security, John Wiley & Sons, 1982.

[Ra]: Raghavendran, R. "Finite Associative Rings", *Compositio Mathematica*, Vol 21, Fasc. 2 (1969), pp. 195-229.

[Ro]: Roby, Norbert, "Sur le Cardinal du Group GL(n,A) où A est un Anneau Fini", *Anais Acad. Brazil C.*, 49 (1977). pp 15-18.

[To]: Toskey, B.R., "Rings on a Direct Sum of Cyclic Groups", *Publ Math Debrecen*, 10 (1963), pp. 93-95.

[W1]: Wiesenbauer, J. "Über die endlichen Ringe mit gegeben additiver Gruppe", *Monatsch. Math.* 78 (1974), pp. 164-173.

[W2]: ________, "Über die endlichen p-Ringe vom Rang Zwei", *Math. Balk.* 46 (1974), pp.723-725.

[W3]: ________ and Walter Flor, "Zum Klassificationsproblem endlicher Ringe", Osterreich Akad. Wiss.der Math-Natur Kl, Abt II, 183 no.8-10, pp.309-320.

Abstract

KIELE, WILLIAM ALBERT. The Classification Problem of Finite Rings by Computable Means (Under the Direction of Kwangil Koh).

The purpose of this paper is to establish a constructive method for testing when two given finite rings are isomorphic. Currently published theory has classified a significant number of finite rings; however, "idealized" representatives are almost always used, with no provision for determining which isomorphism class an arbitrary ring belongs. The new results are as follows:

1. Two rings are isomorphic if and only if a specific system of quadratic equations is satisfied. This system, and a method of attacking it, were developed by the author.
2. As a corollary to the preceding result, there exists a system of linear equations that positively identify whether or not a ring R possesses a 1. The system also shows how to change a ring's basis so that 1 becomes a basis element. Some tests for existence of other idempotents besides 1 are shown.
3. Some old and new results in classifying finite rings of small rank are obtained with the help of theory developed in this paper.

Acknowledgments

No success is ever the sole property of the achiever; I gratefully cite those who contributed most significantly to this milestone in my life.

First and foremost, I thank my Lord and Savior, Jesus Christ, who changed my life 15 years ago and made me a child of God. Without the Lord's guidance, much of this life would be much less meaningful; I thank Him for helping me keep things in His eternal perspective.

Second, I thank my wife, Tracy, who has provided continuing love, devotion, and support. Because of her sacrifice, our family has thrived during our thirteen happy years of marriage.

Professor Koh has played the key role for the last two years in my studies; when disappointment was encountered, he provided much-needed encouragement; his questions steered me toward new ways of thinking about my research problem; most importantly, he was always available to discuss this research with me. I could not have had a better committee chairman, and I am grateful our paths crossed.

I asked Professors Martin, Singer, and Stitzinger to serve on my committee because I respect them not only academically, but also personally. I thank them for their ideas, their impact on my study habits, and their time. Also, a special thanks to Dr. Franke for keeping me on schedule and knowledgeable of institutional requirements.

Finally, I am grateful to the Air Force Academy Department of Mathematical Sciences for their willingness to sponsor my pursuit of this doctorate and for their flexibility during the early days, when my initial course of study proved to be "wrong" for me.

Table of Contents

Introduction

Raghavendran [Ra], Wiesenbauer [W1],[W2], and Kruse & Price [KP] classified all finite p-rings of cardinality up to and including p^4, using well-established theory of Jacobsen radicals and innovative techniques of their own. In each work, these authors also tackled special larger rings using their methods. In all cases, the "canonical" forms each author used were carefully chosen for their simplicity of expression and invariance under various algebraic properties. Further, key ring isomorphism theorems and results were based on existence, rather than construction. In this paper, an approach to the ring isomorphism problem is taken which is related to both Wiesenbauer's and Kruse & Price's; yet, the differences are sufficient to yield new insights so that one can see <u>how</u> two rings of small rank are related to each other, by actually constructing a transition mapping. The chapters of this paper are organized as follows:

I. Two familiar objects are looked at in a new way--the multiplication table of a ring defined on the basis of its additive group, and its associated coordinate "cube". While Wiesenbauer and Kruse & Price both define structure constants based on at least one of the axes of the cube, none look at all three. By so doing, some new computational results equivalent to and extending the currently known ones are obtained.

II. A mapping on the module of cubes is defined and shown to be a group representation; other properties are also obtained.

III. A conclusive test for isomorphism between two rings is derived, producing a system of quadratic equations which can be effectively attacked when the rank of R is small. This system, called the Quadratic Identities, can be modified to identify any and all nonzero idempotents in a given ring, including 1.

IV. A presentation of the computer algorithms developed for this paper, as well as several illustrative examples, are given.

V. Some old and new results for certain rings of rank two are obtained, using idempotents and the tools of this paper as the basis for investigation.

I. Review and Elementary Results

1. Two classical theorems in group theory.

Convention. Throughout this paper, a ring will be presumed to be associative except where noted, and not necessarily with identity. Also, R^+ means the additive group of a ring, or any abelian group from which a ring is to be constructed --the context will make it clear which of the two is meant.

Proposition I.1.1: Let R be a finite ring. Then R is the (ring) direct sum $R \cong \overset{n}{\oplus} R_i$, $1 \le i \le t$, where $|R_i| = p_i^{\alpha_i}$, each p_i is a distinct prime number, and each α_i is a positive integer.

Proof: Since R is a finite ring, $|R| = p_1^{\alpha_1} \cdot \ldots \cdot p_t^{\alpha_t}$ and it is an abelian group (denoted R^+) with respect to its addition. Thus, R^+ is the additive direct sum of its (unique) Sylow p_i-subgroups. It must be shown that with respect to the ring multiplication, the Sylow groups are two-sided ideals as well.

Let R_i denote the Sylow p_i-subgroup of R^+. Then $|R_i| = p_i^{\alpha_i}$. Further, if $x \in R_i$, then $p_i^k \cdot x = 0$ for some nonnegative integer k, because each element of R_i generates a subgroup of R_i; hence, o(x), the <u>additive</u> order of x,

divides $p_i^{d_i}$. Hence

$$R_i \leq R_i' = \{x \in R \mid p_i^k \cdot x = 0 \text{ for some } k \geq 0\}.$$

The sets are in fact equal, for any element in R_i' generates a p_i-group which can be a subgroup only of R_i, and no other R_j, because the order of a subgroup divides the order of the group.

Now R_i' is clearly a ring, and further, since for all $x \in R$, $x \cdot R_i' \leq R_i'$ and $R_i' \cdot x \leq R_i'$, it is a two-sided ideal. The desired result is thus proven. ■

Thus, when considering finite rings, it is sufficient to examine finite rings of prime-power order. Such rings will be called p-rings. Throughout this paper, additive notation will be used to describe the operation in an abelian group.

The following theorem is known, and a constructive proof can be found in [Ca].

Theorem I.1: Every finite abelian p-group G is the direct sum of cyclic subgroups.

Remark 1.1.1: Thus, if G is a finite abelian p-group, then G is isomorphic to $\mathbb{Z}/p^{d_1} \oplus \mathbb{Z}/p^{d_2} \oplus \ldots \oplus \mathbb{Z}/p^{d_k}$, and the

following facts are a consequence of Theorem I.1:

a. The numbers $d_1,..,d_k$ are unique up to rearrangement.

b. Any subgroup generated by an element of maximal order is a direct summand.

c. If a ring R has a 1, then since 1 is a maximal order element additively, <1> is a direct summand.

Definition I.1.1: A basis for an abelian group R^+ is any minimal set of generators of R^+. When the elements of the basis are placed in nonincreasing additive order, they will be said to be in natural order.

Definition I.1.2: The rank of R^+ (and hence of R) is the number of elements in a basis for R^+. The type of R is the k-tuple $(d_1,..,d_k)$, and will be denoted $_p(d_1,..,d_k)$.

Remarks:

1.1.2. The term "basis" is justified because every element $x \in R$ has a unique representation $x = c_1e_1+..+c_ke_k$ for $c_i \in \mathbb{Z}/p^{d_i}$.

1.1.3. If $o(e_i) = p^d$ for all i=1,..,k, then G is a free $\mathbb{Z}/p^d$-module. Otherwise, G is the direct sum of free $\mathbb{Z}/p^{d_i}$-modules, where $d_1>d_2>..>d_n>0$, $n \le k$. Such a ring

will be called a <u>mixed-order</u> ring. Mixed-order rings can be thought of as a free $\mathbb{Z}/p^{d_1}$-module of rank k, factored by the following equivalence relation:

Proposition I.1.2: Let $x, y \in R^+$, $\mathcal{B} = \{e_i\}_{i=1}^{k}$ a basis for R^+ in natural order, $x = \sum c_i e_i$, $y = \sum d_i e_i$, $c_i, d_i \in \mathbb{Z}/p^{d_1}$. Then

$$x = y \text{ if and only if } c_i \equiv d_i \bmod p^{d_i} \text{ for all } i = 1,..,k.$$

Proof: $x-y = \sum_{i=1}^{k} (c_i - d_i)e_i$. $x = y \Leftrightarrow x-y = 0_R \Leftrightarrow$ $\sum_{i=1}^{k} (c_i - d_i)e_i = 0_R \Leftrightarrow c_i - d_i \equiv 0 \bmod p^{d_i}$ for all i=1,.,k since $\mathcal{B}$ is a basis. ■

The foundations of ring structure will now be introduced by means of special mappings known as multiplications.

Definition I.1.3: A <u>multiplication</u> on a group G is a mapping $\mu: G \times G \longrightarrow G$ which is bilinear with respect to the group operation. μ is <u>associative</u> provided $\mu(\mu(a,b),c) = \mu(a,\mu(b,c))$ for all $a,b,c \in R$.

Theorem I.2 ([Fu]): Every multiplication on an abelian group G can be characterized by its action on any basis for G. Conversely, any mapping defined on a basis of G,

subject to the condition $o(\mu(\mathbf{a},\mathbf{b})) \leq \min\{o(\mathbf{a}),o(\mathbf{b})\}$, extends to a multiplication on all of G. Finally, a multiplication is (commutative, associative) over all of G if and only if it is (commutative, associative) over its basis.

Note: Each multiplication defines a (not necessarily associative) ring when G is abelian. Fuchs points out that for any group G the set Mult G = $\{\mu$: μ is a multiplication on G$\}$ is a group; its group operation is given $\mu \vee \nu(\mathbf{a},\mathbf{b}) = \mu(\mathbf{a},\mathbf{b})+\nu(\mathbf{a},\mathbf{b})$ ("+" is the operation on G). Mult G is abelian when G is. He also notes that the set of associative multiplications is a subset, and normally not a subgroup, of Mult G.

Example I.1.1: Let $G = \mathbb{Z}/p \oplus \mathbb{Z}/p$. Let $\mu_1[(\mathbf{a}_1,\mathbf{b}_1),(\mathbf{a}_2,\mathbf{b}_2)] = (\mathbf{a}_1\mathbf{a}_2,\mathbf{b}_2)$. Let $\mu_2[(\mathbf{a}_1,\mathbf{b}_1),(\mathbf{a}_2,\mathbf{b}_2)] = (\mathbf{a}_1,\mathbf{b}_2)$. Both μ_1 and μ_2 are associative by direct calculation. However, $(\mu_1 \vee \mu_2)$ is not associative.□

2. Introduction of basic "machinery" for ring computations.

Definition I.2.1: Given R^+ an abelian group with basis $\mathcal{B} = \{e_1, \ldots, e_k\}$, let

$$M = \begin{array}{c|ccc} \mu & e_1 & \cdots & e_k \\ \hline e_1 & \mu(e_1,e_1) & \cdots & \mu(e_1,e_k) \\ \vdots & \vdots & & \vdots \\ e_k & \mu(e_k,e_1) & \cdots & \mu(e_k,e_k) \end{array} .$$

M is called the <u>multiplication table on G with respect to the basis</u> $\mathcal{B}$ (denoted $M_{\mathcal{B}}$ when basis identification is necessary).

For brevity of notation, the symbol $\mathbb{Z}_p[x]$ will frequently be used to denote $\mathbb{Z}[x]/_p$.

Example I.2.1: Let $R = GF(3^3)$, the field of 27 elements. One representation of R is $\mathbb{Z}_3[x]/_{(x^3+2x+1)}$. A basis for R^+ is $\mathcal{B} = \{1, x, x^2\}$. The multiplication table of R with respect to $\mathcal{B}$ is:

$$M = \begin{array}{c|ccc} \mu & 1 & x & x^2 \\ \hline 1 & 1 & x & x^2 \\ x & x & x^2 & x+2 \\ x^2 & x^2 & x+2 & x^2+2x \end{array}$$

Remarks:

1.2.1. The multiplication table is crucial to all the theoretical development in this paper, since all the basic computational results flow from its analysis.

1.1.2. Throughout this paper, bold letters such as **e** and **f** will be reserved for ring elements, and normally-typed letters will denote scalars. Further, $\mathbf{e}_{ij}$ will mean $\mu(\mathbf{e}_i,\mathbf{e}_j)$. No confusion should result from this choice of notation.

As $\mathcal{B} = \{\mathbf{e}_1,..,\mathbf{e}_k\}$ serves as a coordinate system for G, one can coordinatize the basis multiplication table for $R = (R^+,\mu)$.

Definition I.2.2: The <u>coordinate cube</u> (or simply a <u>cube</u>) of $R = (R^+,\mu)$ is the k×k×k array of coordinates for each element $\mathbf{e}_{ij} = \mu(\mathbf{e}_i,\mathbf{e}_j)$ of the multiplication table M. It will be denoted by [M] (or $[M]_{\mathcal{B}}$ when context requires basis identification), and the standard display will be

$$M = \begin{bmatrix} \begin{bmatrix} m^1_{11} \\ \vdots \\ m^k_{11} \end{bmatrix} & \begin{bmatrix} m^1_{1k} \\ \vdots \\ m^k_{1k} \end{bmatrix} \\ & \\ \begin{bmatrix} m^1_{k1} \\ \vdots \\ m^k_{k1} \end{bmatrix} & \begin{bmatrix} m^1_{kk} \\ \vdots \\ m^k_{kk} \end{bmatrix} \end{bmatrix}, \quad \text{where } \mathbf{e}_{ij} = m^1_{ij}\mathbf{e}_1+..+m^k_{ij}\mathbf{e}_k\,, \quad m^t_{ij} \in \mathbb{Z}/p^{d_t}\,.$$

Further, if $\ell = \min\{d_i,d_j\}$, then $p^{d_t-\ell}$ divides m^t_{ij} when $d_t > \ell$. The reason for this restriction is contained in Proposition I.3.1.

Example I.2.2: Let R^+ be of type ${}_2(4,2)$. Let

$$[M] = \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}8\\1\end{bmatrix} \end{bmatrix} \quad \text{and} \quad [N] = \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}2\\1\end{bmatrix} \end{bmatrix}$$

[M] represents a multiplication on R^+, but [N] doesn't.

Remark 1.2.3: One can view the cube in three dimensions, and then examine its slices along each principal axis. These slices will be used extensively, and so the following notation will be employed (See Figure I.2, below):

$[M]^{\ell}_{**}$ means the ℓ-th horizontal slice of [M].

$[M]^{*}_{i*}$ means the i-th back-to-front slice of [M].

$[M]^{*}_{*j}$ means the j-th left-to-right slice of [M].

μ	e_1	..	e_k
e_1	e_{11}	..	e_{1k}
$\vdots$	$\vdots$		.
e_k	e_{k1}	..	e_{kk}

$\longleftarrow$ a sample multiplication table M

$$e_{ij} = m^1_{ij} e_1 + \ldots + m^k_{ij} e_k \quad \longleftrightarrow$$

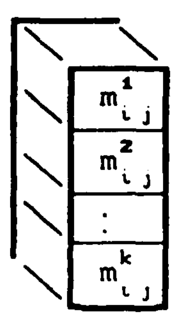


Figure I.1. A "picture" of a coordinate stack.

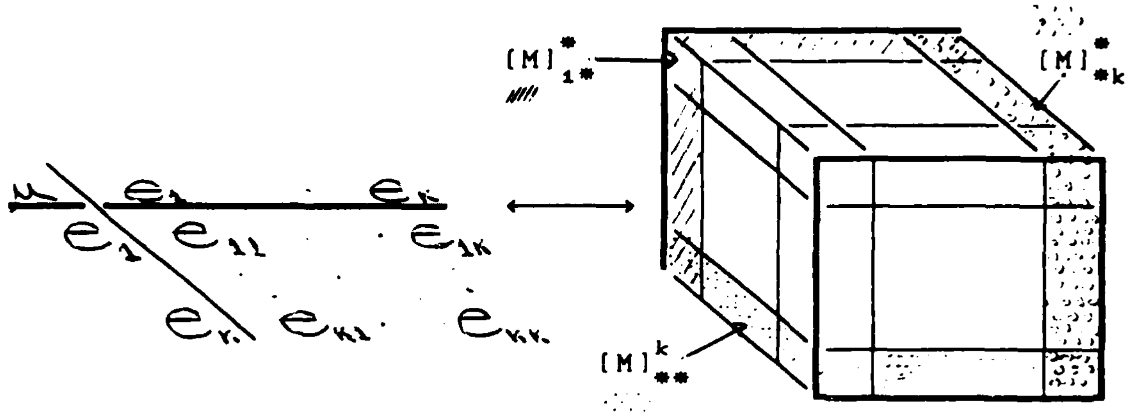


Figure I.2. A "picture" of a coordinate cube.

If two rings R and R' are to be isomorphic, they must have isomorphic additive groups R^+ and R'^+. Thus without loss of generality, we can set $R^+ = R'^+$.

Definition I.2.3: We say two multiplications μ and ν are <u>equivalent</u> provided $(R^+,\mu) \cong (R^+,\nu)$.

Example I.2.3: Let $R = \mathbb{Z}_3[x]/_{(x^3+2x+1)}$ and

$R' = \mathbb{Z}_3[x]/_{(x^3+2x^2+1)}$. They both represent $GF(3^3)$, since both polynomials are irreducible mod 3. M and M' are, respectively,

μ	1	x	x^2
1	1	x	x^2
x	x	x^2	$x+2$
x^2	x^2	$x+2$	x^2+2x

ν	1	x	x^2
1	1	x	x^2
x	x	x^2	x^2+2
x^2	x^2	x^2+2	x^2+2x+2

By inspection, μ and ν are different multiplications, but they are equivalent, since all fields with the same number of elements are known to be isomorphic.□

It is clear that equivalence of multiplications is an equivalence relation, and that Mult G is decomposed by this relation into disjoint sets. The following results describe how the decomposition is accomplished.

First, some results concerning bases of R^+:

Proposition I.2.1: Given R^+, let $\mathcal{B}_1 = \{e_i\}_{i=1}^k$ and $\mathcal{B}_2 = \{f_i\}_{i=1}^k$ be two bases for R^+. Then there exists a matrix $A \in GL(k, \mathbb{Z}/p^d)$, $d = \max\{d_i\}$, such that

$[e_1 \,..\, e_k]\cdot A = [f_1 \,..\, f_k]$ ($\cdot$ means matrix multiplication)

Proof: Since $\mathcal{B}_1$ is a basis, the f_i can be expressed by:

$$\left.\begin{matrix} f_1 = a_{11}e_1+\dots+a_{k1}e_k \\ \vdots \\ f_k = a_{1k}e_1+\dots+a_{kk}e_k \end{matrix}\right\} \Rightarrow [e_1..e_k]\cdot\begin{bmatrix} a_{11} & .. & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & .. & a_{kk} \end{bmatrix} = [f_1..f_k]. \tag{I.2.1}$$

Denote $[a_{ij}]$ by A.

Similarly, since $\mathcal{B}_2$ is also a basis, the e_i can be expressed by:

$$\left.\begin{array}{l} \mathbf{e}_1 = b_{11}\mathbf{f}_1+\ldots+b_{k1}\mathbf{f}_k \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \\ \mathbf{e}_k = b_{1k}\mathbf{f}_1+\ldots+b_{kk}\mathbf{f}_k \end{array}\right\} \Rightarrow [\mathbf{f}_1..\mathbf{f}_k]\cdot\begin{bmatrix} b_{11} & .. & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{k1} & .. & b_{kk} \end{bmatrix} = [\mathbf{e}_1..\mathbf{e}_k]. \tag{I.2.2}$$

Therefore $[\mathbf{f}_1\,..\mathbf{f}_k]\cdot[b_{ij}]\cdot A = [\mathbf{f}_1\,..\mathbf{f}_k]$, hence $[b_{ij}]\cdot A \equiv I \bmod p^d$. Since $A\cdot(\mathrm{adj}\ A) = |A|\cdot I$, we conclude that $|A|$ is a unit mod p^d, hence A is nonsingular. ∎

The converse of the last proposition may not be true: i.e., certain elements of $GL(k,\mathbb{Z}/_{p^d})$ acting on a basis $\mathcal{B}$ may not produce another basis.

Example I.2.4: Consider R of type $_p(4,2)$, and let

$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. A is nonsingular, but the set $\{\mathbf{e}_1, \mathbf{e}_1 + \mathbf{e}_2\}$ is not a basis, because the additive order of $\mathbf{e}_1 + \mathbf{e}_2$ is p^4 and not p^2.

However, by restricting one's attention to a subset of $GL(k,\mathbb{Z}/_{p^d})$, a converse of sorts can be established. But first, a lemma:

Lemma: If $\mathcal{B} = \{\mathbf{e}_1,..,\mathbf{e}_k\}$ is a basis for R^+, then $\mathcal{B}' = \{\mathbf{e}_1,..,\mathbf{e}_{j-1},(a_{1j}\mathbf{e}_1+..+a_{kj}\mathbf{e}_k),\mathbf{e}_{j+1},..,\mathbf{e}_k\}$ is a basis

for R^+ if and only if $(a_{jj},p) = 1$, $o(\sum_i a_{ij}\mathbf{e}_i) = o(\mathbf{e}_j)$.

Proof: $\Rightarrow$: If $\mathcal{B}'$ is a basis for R^+, then

$o(\sum_i a_{ij}\mathbf{e}_i) = o(\mathbf{e}_j) = p^{d_j}$ because of the invariant class property of abelian groups. Further, $(a_{jj},p) = 1$, for if not, the basis change matrix, which is

$$A = \begin{bmatrix} 1 & a_{1j} & \cdots & 0 \\ 0 & \cdot & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{kj} & \cdots & 1 \end{bmatrix}$$ would be singular mod p,

contradicting the previous Proposition.□

$\Leftarrow$: Assume the converse. All that is necessary to be shown is that the j-th element of $\mathcal{B}'$ is independent of the other elements of the set. Then $\mathcal{B}'$ will be a basis for R^+ because the orders of the elements of $\mathcal{B}$ and $\mathcal{B}'$ coincide.

Let $c_1\mathbf{e}_1+..+c_{j-1}\mathbf{e}_{j-1}+c_j(\sum_{i=1}^{k} a_{ij}\mathbf{e}_i)+..+c_k\mathbf{e}_k = \mathbf{0}_R$.

Then $(c_1+c_j a_{1j})\mathbf{e}_1+..+c_j a_{jj}\mathbf{e}_j+..+(c_j a_{kj}+c_k)\mathbf{e}_k= \mathbf{0}_R$. (I.2.3)

$\mathcal{B}$ is a basis, hence $\begin{cases} c_i+c_j a_{ij} \equiv 0 \bmod p^{d_i}, & j\neq i \\ c_j a_{jj} \equiv 0 \bmod p^{d_j} \end{cases}$.

$(a_{jj},p) = 1 \Rightarrow a_{jj}^{-1}$ exists $\Rightarrow c_j = p^{d_j}\cdot d$ for some $d \in \mathbb{Z}$. Thus we have two cases:

1. $d_i \leq d_j$. Then $c_i \equiv 0 \bmod p^{d_i}$.

2. $d_i > d_j$. Then $c_i = p^{d_j}[p^{d_i-d_j}\cdot\gamma - d\cdot a_{ij}]$ for some

$\gamma, d \in \mathbb{Z}$. $o(\sum_{i=1}^{k} a_{ij} e_i) = o(e_j) \Rightarrow p^{d_i - d_j} | a_{ij}$ for all $d_i > d_j$.

Therefore $c_i = p^{d_i}[\gamma - d \cdot b_{ij}]$ for some $\gamma, d, b_{ij} \in \mathbb{Z}$. Thus we have $c_1, \ldots, c_k \equiv 0$. ■

Proposition I.2.2: Let $\mathcal{B}_1$ be a basis for R^+, $A \in GL(k, \mathbb{Z}/_{p}d)$ with the following restriction:

$p^{d_i - d_j} | a_{ij}$ for all $d_i > d_j$, $a_{ij} \in \mathbb{Z}/_{p}d_i$.

Then $\mathcal{B}_2 = \{\sum_j a_{j1} e_j, \ldots, \sum_j a_{jk} e_j\}$ is also a basis for R^+.

Proof: Repeated application of Lemma. Here's how: Because A is nonsingular mod p, $|A|$ is a unit mod p $\Rightarrow$ there exists $\sigma \in S_k$ (the permutation group of k letters) such that $a_{1\sigma(1)} \cdots a_{k\sigma(k)}$ is a unit mod p. Rearrange $\mathcal{B}_1$ to be $\{e_{\sigma(1)}, \ldots, e_{\sigma(k)}\}$. Let $\mathcal{B}_2 = \{\sum a_{i\sigma(1)} e_i, e_{\sigma(2)}, \ldots, e_{\sigma(k)}\}$. The conditions of the Lemma are now satisfied, and so $\mathcal{B}_2$ is a basis. Continuing the chain of transitions, the proposition now follows. ■

Notice that in the proof of the proposition, A had to be restricted in the case of rings of mixed-order. In free $\mathbb{Z}/_{p}d$ - modules, this isn't necessary, and in this case, any matrix $A \in GL(k, \mathbb{Z}/_{p}d)$ will transform one basis into

another. In order to make clear what the restriction means, assume that a mixed-order ring R has two bases arranged in natural order (something not necessary to the proof of the previous propositions). Then the transition matrix would satisfy the following equation:

$[\mathbf{e}_1 \;..\; \mathbf{e}_k]\cdot A = [\mathbf{f}_1 \;..\; \mathbf{f}_k]$ ($\cdot$ means matrix multiplication).

Because $o(\sum_i a_{ij}\mathbf{e}_i) = o(\mathbf{f}_j) = p^{d_j}$, we have

$p^{d_i - d_j} | a_{ij}$ for all $i < j$, since $o(\mathbf{e}_i) \geq o(\mathbf{e}_{i+1})$. The basis change matrix, then, has an upper triangle restricted to elements divisible by appropriate powers of p. Thus the set of basis change matrices for any mixed-order ring R is a proper subset of $GL(k, \mathbb{Z}/p^{d})$. This is the motivation for the following definition:

Definition I.2.4: The set of all basis change matrices for a particular group R^+ will be known as the set of Transition Matrices of the group R^+. It will be denoted by $\mathrm{Tran}\,(d_i)_{i=1}^{k}$. Specifically, $\mathrm{Tran}\,(d_i)_{i=1}^{k}$ =

$$\{A \in GL(k, \mathbb{Z}/p^{d}) \mid d = \max\{d_i\},\ p^{d_{\sigma(i)} - d_{\sigma(j)}} | a_{\sigma(i)\sigma(j)}\},$$

where $\sigma \in S_k$, the symmetric group of k letters and $d_{\sigma(i)} > d_{\sigma(j)}$.

It is not necessary to restrict $a_{ij} \in \mathbb{Z}/p^{\alpha_i}$, because of the following proposition:

Proposition I.2.3: Let $\mathcal{B}$ be a basis for R^+, $A_1, A_2 \in \mathrm{Tran}\ (\alpha_i)_{i=1}^k$. Then

$[e_1..e_k]\cdot A_1 \equiv [e_1..e_k]\cdot A_2$ if and only if $[A_1]_{i*} \equiv [A_2]_{i*} \bmod p^{\alpha_i}$

for all i=1,..,k.

Proof: A straightforward computation. ■

Proposition I.2.3 implies that one can think of a "unique mod p^{α_i}" transition matrix between two bases for R^+.

Proposition I.2.4: Tran $(\alpha_i)_{i=1}^k$ is a subgroup of $GL(k, \mathbb{Z}/p^{\alpha})$.

Proof: It is only necessary to consider the mixed-order ring case. Must show that if $A_1, A_2 \in \mathrm{Tran}\ (\alpha_i)_{i=1}^k$,then $A_1 \cdot A_2^{-1}$ is in Tran $(\alpha_i)_{i=1}^k$. That A_2^{-1} exists in Tran $(\alpha_i)_{i=1}^k$ is clear, for it is the transition mapping from $A_2(\mathcal{B})$ back to $\mathcal{B}$. The composition of two changes of basis, which is the product of two transition matrices, is a change of basis, and thus is a transition matrix. This completes the proof.■

The following theorem is well-known. It is expressed here in a way consistent with the notation developed thus far.

Theorem I.3 [Fu]: Given $R_1=(R^+,\mu)$, $R_2=(R^+,\nu)$ two finite p-rings. Then $R_1 \cong R_2$ if and only if there exist two bases, $\mathcal{B}$ and $\mathcal{B}'$, for R_1 and R_2 respectively, such that $[M]_B=[N]_{B'}$.

Proof: Some preliminaries for use in this theorem are necessary:

Let $\mathcal{B} = \{e_1,..,e_k\}$. Then

$$M_B = \begin{array}{c|ccc} \mu & e_1 & \cdots & e_k \\ \hline e_1 & e_{11} & \cdots & e_{1k} \\ \vdots & \vdots & & \vdots \\ e_k & e_{k1} & \cdots & e_{kk} \end{array} \longleftrightarrow [M]_B = \begin{bmatrix} [m^t_{11}] & \cdots & [m^t_{1k}] \\ \vdots & \ddots & \vdots \\ [m^t_{k1}] & \cdots & [m^t_{kk}] \end{bmatrix}_B \quad t=1,..,k.$$

$\Rightarrow$: Suppose $R \cong R'$. Then there exists a ring isomorphism $\varphi:R\longrightarrow R'$ such that $\varphi(\mathbf{a}+\mathbf{b}) = \varphi(\mathbf{a})+\varphi(\mathbf{b})$ and $\varphi(\mu(\mathbf{a},\mathbf{b})) = \nu(\varphi(\mathbf{a}),\varphi(\mathbf{b}))$ for all $\mathbf{a},\mathbf{b} \in R$.

It is straightforward to show that $\mathcal{B}'=\{\varphi(e_1),..,\varphi(e_k)\}$ is also a basis. Now the following tables are identical:

$$N_{B'} = \begin{array}{c|ccc} \nu & \varphi(e_1) & \cdots & \varphi(e_k) \\ \hline \varphi(e_1) & \nu(\varphi(e_1),\varphi(e_1)) & \cdots & \nu(\varphi(e_1),\varphi(e_k)) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(e_k) & \nu(\varphi(e_k),\varphi(e_1)) & \cdots & \nu(\varphi(e_k),\varphi(e_k)) \end{array}$$

and

$$\begin{array}{c|ccc} \nu & \varphi(e_1) & \ldots & \varphi(e_k) \\ \hline \varphi(e_1) & \varphi(e_{11}) & \ldots & \varphi(e_{1k}) \\ \vdots & & & \vdots \\ \varphi(e_k) & \varphi(e_{k1}) & \ldots & \varphi(e_{kk}) \end{array}$$

Also,

$\varphi(e_{ij}) = \varphi(m_{ij}^1 e_1 + .. + m_{ij}^k e_k) = m_{ij}^1 \varphi(e_1) + .. + m_{ij}^k \varphi(e_k)$.

Hence $[N]_{\mathcal{B}'} = [M]_{\mathcal{B}}$.□

⇐: Assume the converse. Put $\mathcal{B}$ and $\mathcal{B}'$ in natural order and

let $A = \begin{bmatrix} a_{11} \ldots a_{1k} \\ \vdots \\ a_{k1} \ldots a_{kk} \end{bmatrix}$ be the basis change matrix between the

bases $\mathcal{B}$ and $\mathcal{B}'$. Define $\varphi : \mathcal{B} \longrightarrow \mathcal{B}'$by

$\varphi(e_i) = \sum_j a_{ji} e_j = f_i \in \mathcal{B}'$. As φ is defined on the basis $\mathcal{B}$,

one can naturally extend φ to $\Phi : R_1 \longrightarrow R_2$ by

$\Phi(a) = \Phi(c_1 e_1 + .. + c_k e_k) \stackrel{\Delta}{=} c_1 \varphi(e_1) + .. c_k \varphi(e_k)$. The proof that

Φ is an isomorphism is a straightforward calculation.■

3. Some computationally obtained properties of a ring.

This first Proposition was known to Baumgartner [Ba] and Toskey [To] in other guises.

Proposition I.3.1: Let $\mathcal{B} = \{e_1,..,e_k\}$ be a basis for R^+, $o(e_i)=p^{d_i}$. Let [M] be a coordinate cube over $\mathbb{Z}/_p d$, $d = \max\{d_i\}$. Then [M] represents a (not necessarily associative) multiplication on $\mathcal{B}$ if and only if

$\min\{p^{d_r},p^{d_i},p^{d_j}\}\cdot m^r_{ij} \equiv 0 \bmod p^{d_r}$.

Proof: This is another way of expressing Theorem I.2. To see this, note that $o(e_{ij}) \le \min\{o(e_i),o(e_j)\} \Leftrightarrow$ $\min\{p^{d_i},p^{d_j}\}\cdot e_{ij} = 0_R \Leftrightarrow \min\{p^{d_i},p^{d_j},p^{d_r}\}\cdot m^r_{ij} \equiv 0 \bmod p^{d_r}$, $r=1,..,k$. Using Theorem I.2, the Proposition now follows.■

Corollary I.3.1a: Given $\mathcal{B}$ and [M] as above, $\mathcal{B}$ in natural order. Then [M] represents a multiplication on the basis $\mathcal{B}$ if and only if $p^{d_r-d_j}$ divides m^r_{ij} for all $i,r \le j$, and $p^{d_r-d_i}$ divides m^r_{ij} for all $j,r \le i$.

In other words, the slices $[M]^*_{i*}$ and $[M]^*_{*j}$ have the same upper triangular structure as do transition matrices. Of course, they need not be nonsingular.

Corollary I.3.1b: If R is a free $\mathbb{Z}/_{p}d$-module, then every cube represents a multiplication on $\mathcal{B}$.

Proof: If R is a free $\mathbb{Z}/_{p}d$-module, then R is of type $_p(d,d,..d)$, and thus by the proposition, no restriction on [M] is imposed.■

Proposition I.3.2: Given the same conditions in Proposition I.3.1, [M] represents an associative ring if and only if $[M]^*_{i*}\cdot[M]^*_{*t} = [M]^*_{*t}\cdot[M]^*_{i*}$ for all i,t = 1,..,k.
(This above result was shown in [W1], but this proof is more elementary.)

Proof: By definition (and Theorem I.2), a multiplication table M is associative provided
$(e_{ij})e_t = e_i(e_{jt})$ for all i,j,t=1,..,k. This means that

$$m^1_{ij}e_{1t}+..+m^k_{ij}e_{kt} = m^1_{jt}e_{i1}+..+m^k_{jt}e_{ik} \text{ for all } i,j,t=1,..,k. \tag{I.3.1}$$

$$\Rightarrow m^1_{ij}\sum_s m^s_{1t}e_s +..+m^t_{ij}\sum_s m^s_{tt}e_s = m^1_{jt}\sum_s m^s_{i1}e_s+..+m^t_{jt}\sum_s m^s_{it}e_s. \tag{I.3.2}$$

Regrouping terms, we get

$$\sum_u m^u_{ij}m^1_{ut}e_1 +..+\sum_u m^u_{ij}m^k_{ut}e_k = \sum_u m^u_{jt}m^1_{iv}e_1 +..+\sum_u m^u_{jt}m^t_{iv}e_t. \tag{I.3.3}$$

As both sides of the equations are inner products, one obtains

$$\left[\sum_u m_{ij}^u m_{ut}^1 \;\cdots\; \sum_u m_{ij}^u m_{ut}^k\right] \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix} = \left[\sum_u m_{jt}^u m_{iu}^1 \;\cdots\; \sum_u m_{jt}^u m_{iu}^k\right] \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix},$$

Which in turn gives

$$\left[m_{ij}^1 \;\cdots\; m_{ij}^k\right] \cdot \begin{bmatrix} m_{1t}^1 & \cdots & m_{1t}^k \\ \vdots & & \\ m_{kt}^1 & & m_{kt}^k \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix}$$

$$= \left[m_{jt}^1 \;\cdots\; m_{jt}^k\right] \cdot \begin{bmatrix} m_{i1}^1 & \cdots & m_{i1}^k \\ \vdots & & \\ m_{ik}^1 & & m_{ik}^k \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix}.$$

Since this identity holds for all i,j,t, the following matrix identities also hold, letting j range from 1 to k:

$$\begin{bmatrix} m_{i1}^1 & \cdots & m_{i1}^k \\ \vdots & & \\ m_{ik}^1 & & m_{ik}^k \end{bmatrix} \cdot \begin{bmatrix} m_{1t}^1 & \cdots & m_{1t}^k \\ \vdots & & \\ m_{kt}^1 & & m_{kt}^k \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix}$$

$$= \begin{bmatrix} m_{1t}^1 & \cdots & m_{1t}^k \\ \vdots & & \\ m_{kt}^1 & & m_{kt}^k \end{bmatrix} \cdot \begin{bmatrix} m_{i1}^1 & \cdots & m_{i1}^k \\ \vdots & & \\ m_{ik}^1 & & m_{ik}^k \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ \vdots \\ e_k \end{bmatrix},$$

which one can see is

$[e_1 \;..\; e_k]\cdot[M]^*_{*t}\cdot[M]^*_{i*} = [e_1 \;..\; e_k]\cdot[M]^*_{i*}\cdot[M]^*_{*t}$.

Thus the Proposition is proven.■

Proposition I.3.3: A ring R is commutative if and only if its cube has the property that $[M]^r_{**}$ is symmetric for all r=1,..,k.

(**[W1]** gives an equivalent, though "longer" test; namely,

that $[M]^{*}_{i*} = [M]^{*}_{*i}$ for all $i=1,..,k$.

Proof: Again using Theorem I.2, a ring is commutative if and only if its multiplication table is; i.e. if and only if $\mathbf{e}_{ij} = \mathbf{e}_{ji}$ for all $i,j=1,..,k$, $i \neq j$.

The result immediately follows by examining the slices $[M]^{r}_{**}$, which are the r-th coordinates of the $\mathbf{e}_{ij}$. ∎

The significance of this next proposition is that once the i-th row and column of one cube representing R are made to agree with the i-th row and column of another cube representing R, then the corresponding basis elements $\mathbf{e}_i$ and $\mathbf{f}_i$ of $\mathcal{B}_1$ and $\mathcal{B}_2$, repsectively, "behave" the same, and the basis change matrix should not modify them.

Proposition I.3.4: Given two cubes [M] and [N] such that $[M]^{*}_{i*} = [N]^{*}_{i*}$, $[M]^{*}_{*i} = [N]^{*}_{*i}$ for some $i = 1,..,k$.
Then column i of the transition matrix **A** is column i of $\mathbf{I}_k$, the k×k identity matrix.

There are two other very important properties of a ring which must await machinery to be derived in Chapter III; namely, when does a ring have a **1**, and when does a ring have a central idempotent other than a **1** or **0**?

II. RING ISOMORPHISMS AS A GROUP ACTION

1. Basic Terminology and Results.

The cube as a mathematical object has merit in its own right, not just as a potential representative of a finite ring. In this chapter, the cube will be presented in its most general form.

Let $\mathfrak{X}$ be the set of all cubes over $\mathbb{Z}/p^{d}$ of size k; i.e., $\mathfrak{X} \triangleq \{[M] = [m^{u}_{ij}] \mid m^{u}_{ij} \in \mathbb{Z}/p^{d}\,,\ i,j,u=1,..,k\}$. One can perform coordinate-wise addition and multiplication mod $\mathbb{Z}/p^{d}$, and thus $\mathfrak{X} \cong \left(\mathbb{Z}/p^{d}\right)^{k^3}$ as a ring and as a $\mathbb{Z}/p^{d}$ - module.

Definition II.1.1: Let $A \in GL(k, \mathbb{Z}/p^{d})$, $A = (a_{ij})$, and $A^{-1} = (\bar{a}_{ij})$. Define $\mathcal{T}_{A} : \mathfrak{X} \longrightarrow \mathfrak{X}$, where

$$[\mathcal{T}_{A}([M])]^{u}_{ij} \triangleq \sum_{r=1}^{k} \sum_{s=1}^{k} \sum_{t=1}^{k} a_{ri}\, a_{sj}\, \bar{a}_{ut}\, m^{t}_{rs} \qquad \text{(II.1.1)}$$

Since $\mathcal{T}_{A}([M]+[N]) = \mathcal{T}_{A}([M]) + \mathcal{T}_{A}([N])$ and $\mathcal{T}_{A}(c\cdot[M]) = c\cdot\mathcal{T}_{A}([M])$, $c \in \mathbb{Z}/p^{d}$, $\mathcal{T}_{A}$ is seen to be a $\mathbb{Z}/p^{d}$ - linear homomorphism and thus has a matrix representation with respect to a (ring) basis for $\mathfrak{X}$.

Viewed as a ring, $\mathfrak{X}$ has a basis

$\mathcal{B} = \{[T^u_{ij}] \mid i,j,u=1,..,k\}$, where the only nonzero entry of $[T^u_{ij}]$ is $m^u_{ij} = 1$. Further,

$$\mathcal{T}_A([M]) = \mathcal{T}_A\Big(\sum_{i,j,u} m^u_{ij}[T^u_{ij}]\Big) = \sum_{i,j,u} m^u_{ij}\cdot\mathcal{T}_A([T^u_{ij}]). \qquad (II.1.2)$$

Therefore $[N]^u_{ij} \triangleq [\mathcal{T}_A([M])]^u_{ij} \triangleq \sum_{r=1}^{k}\sum_{s=1}^{k}\sum_{t=1}^{k} a_{ri}a_{sj}\bar{a}_{ut}m^t_{rs} \Rightarrow$ (II.1.3)

$$\begin{bmatrix} n^1_{11} \\ \vdots \\ \vdots \\ n^k_{kk} \end{bmatrix} = \begin{bmatrix} a_{11}a_{11}\bar{a}_{11} & \cdots & a_{k1}a_{k1}\bar{a}_{1k} \\ \vdots & \ddots & \vdots \\ a_{1k}a_{1k}\bar{a}_{k1} & \cdots & a_{kk}a_{kk}\bar{a}_{kk} \end{bmatrix} \cdot \begin{bmatrix} m^1_{11} \\ \\ \\ m^k_{kk} \end{bmatrix}$$, which is a

system of k^3 equations. The matrix above is the matrix representation of $\mathcal{T}_A$ with respect to the basis $\mathcal{B}$. It will be denoted by $\mathcal{L}(A)$, and will be referred to later in the paper.

Proposition II.1.1: Given $A,B \in GL(k,\mathbb{Z}/_{p^\alpha})$, $[M] \in \mathfrak{X}$. Then $\mathcal{T}_{A\cdot B}([M]) = \mathcal{T}_B(\mathcal{T}_A([M]))$.

Proof: From Definition II.1.1,

$[\mathcal{T}_A([M])]^u_{ij} = [\sum_{r=1}^{k} a_{ri}\cdot(\sum_{s=1}^{k} a_{sj}\cdot(\sum_{t=1}^{k}\bar{a}_{ut}m^t_{rs}))]$, which implies

$$[\mathcal{T}_B(\mathcal{T}_A([M]))]^u_{ij} = [\sum_{v=1}^{k} b_{vi}\cdot(\sum_{w=1}^{k} b_{wj}\cdot(\sum_{x=1}^{k}\bar{b}_{ux}[\mathcal{T}_A([M])]^x_{vw}))] \ , \qquad (II.1.4)$$

$$= [\sum_{v=1}^{k} b_{vi}\cdot(\sum_{w=1}^{k} b_{wj}\cdot(\sum_{x=1}^{k}\bar{b}_{ux}\ [\sum_{r=1}^{k} a_{rv}\cdot(\sum_{s=1}^{k} a_{sw}\cdot(\sum_{t=1}^{k}\bar{a}_{xt}m^t_{rs}))]. \qquad (II.1.5)$$

Now $[\mathcal{T}_{A\cdot B}([M])]^u_{ij} = [\sum_{r=1}^{k} (AB)_{ri} \cdot (\sum_{s=1}^{k} (AB)_{sj} \cdot (\sum_{t=1}^{k} (B^{-1}A^{-1})_{ut} m^t_{rs}))]$

(II.1.6)

$$= [\sum_{r=1}^{k} \cdot (\sum_{v=1}^{k} a_{rv} b_{vi}) \cdot (\sum_{s=1}^{k} \cdot (\sum_{v=1}^{k} a_{sv} b_{vj}) \cdot (\sum_{t=1}^{k} \cdot (\sum_{x=1}^{k} \bar{b}_{ux} \bar{a}_{xt}) m^t_{rs}))].$$

(II.1.7)

Rearranging the terms of (II.1.5) and (II.1.7) produces equality. ■

The previous Proposition shows that $GL(k, \mathbb{Z}/_p d)$ is a group action over $\mathfrak{X}$.

Definition II.1.2: A <u>commutative cube</u> is any cube such that $[M]^r_{**}$ is symmetric for all r=1,..,k.

Observe that the set $\mathfrak{C}$ of commutative cubes is a linear subspace of $\mathfrak{X}$. Also note that by the definition of $\mathcal{T}_A([M])$, if [M] is commutative, so is $\mathcal{T}_A([M])$. Thus $\mathfrak{C}$ is an invariant subspace under the group action $\mathcal{T}_A$. This makes $\mathcal{T}_A$ reducible into a direct sum of transformations, and $\mathcal{L}(A)$ is thus similar to a block diagonal matrix. The author did not explore any further into this area due to the very large matrices involved.

Definition II.1.3: For $[M] \in \mathfrak{X}$, define

$\mathcal{O}([M]) = \{\mathcal{T}_A([M]) \mid A \in GL(k, \mathbb{Z}/_p d)\}$. $\mathcal{O}([M])$ is called the <u>orbit of [M].</u>

It is clear that $\mathfrak{X}$ is partitioned by this action.

Definition II.1.4: For $[M] \in \mathfrak{X}$, define $Stab([M]) = \{A \in GL(k, \mathbb{Z}/_p d) \mid \mathcal{T}_A([M]) = [M]\}$. Stab([M]) is called the <u>stabilizer of [M]</u>.

Definition II.1.5: We say $[M], [N] \in \mathfrak{X}$ are <u>equivalent</u> provided [M] and [N] are in the same orbit.

Proposition II.1.2: Given [M] equivalent to [N], $\mathcal{T}_A([M]) = [N]$. Then $\mathcal{T}_{A \cdot B}([M]) = [N]$ if and only if $B \in Stab([N])$.

Proposition II.1.2 a: *(Alternate)* Given $\mathcal{T}_A([M]) = [N]$. Then $\mathcal{T}_B([M]) = [N]$ if and only if $B^{-1} \cdot A \in Stab([M])$.

Proof: Both are proven using Proposition II.1.1 and the above definitions.■

Remarks:

II.1.1. Two cubes have the same number of elements in their stabilizers when they are equivalent. However, the

converse is in general false.

II.1.2. Equivalent cubes need not even share the same stabilizers. Hence, stabilizers cannot be used to discover how to break down Tran $(\alpha_i)_{i=1}^{k}$ into some partition analagous to the orbits of a cube.

2. The effect of elementary matrices on cubes.

In linear algebra, there are three matrices which perform the three elementary row operations on a system of equations:

1. The permutation matrix S_{ij} which swaps rows i & j. S_{ij} is a symmetric, involutory matrix.

2. The *scaling matrix* $D = diag(a_1,..,a_k)$ *which* multiplies row i by an invertible constant a_i. D is a diagonal (hence symmetric) matrix with obvious inverse.

3. The elimination matrix $E_{a_{ij}}$ which takes row j and adds a_{ij}·row i to it. This matrix is neither symmetric nor involutory; its inverse is $E_{-a_{ij}}$.

Since all matrices in $GL(k, \mathbb{Z}/_{p}\alpha)$ are products of these matrices, and since $\mathcal{T}_A$ is a group action on $\mathfrak{X}$, it would be fruitful to analyze the effects of these elementary matrices on a cube [M].

To facilitate notation, the word "ijk" will denote m_{ij}^{k}

in M, "ij" will denote a_{ij} in A, and "$\overline{ij}$" will denote $\bar{a}_{ij}$ in A^{-1}.

The following propositions are determined by tabulating equation (II.1.1), and hence their proofs are omitted:

Proposition II.2.1: $\mathcal{T}_{s_{ij}}$ swaps rows <u>and</u> columns i and j of [M], and within each coordinate stack swaps entries i and j. The picture follows:

$$\mathcal{T}_{s_{ij}}([M]) = \begin{bmatrix} \begin{bmatrix}11 1\\11j\\11i\\11k\end{bmatrix} & \cdot\cdot & \begin{bmatrix}1j1\\1jj\\1ji\\1jk\end{bmatrix} & \cdot\cdot & \begin{bmatrix}1i1\\1ij\\1ii\\1ik\end{bmatrix} & \cdot\cdot & \begin{bmatrix}1k1\\1kj\\1ki\\1kk\end{bmatrix} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \begin{bmatrix}j11\\j1j\\j1i\\j1k\end{bmatrix} & \cdot\cdot & \begin{bmatrix}jj1\\jjj\\jji\\jjk\end{bmatrix} & \cdot\cdot & \begin{bmatrix}ji1\\jij\\jii\\jik\end{bmatrix} & \cdot\cdot & \begin{bmatrix}jk1\\jkj\\jki\\jkk\end{bmatrix} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \begin{bmatrix}i11\\i1j\\i1i\\i1k\end{bmatrix} & \cdot\cdot & \begin{bmatrix}ij1\\ijj\\iji\\ijk\end{bmatrix} & \cdot\cdot & \begin{bmatrix}ii1\\iij\\iii\\iik\end{bmatrix} & \cdot\cdot & \begin{bmatrix}ik1\\ikj\\iki\\ikk\end{bmatrix} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \begin{bmatrix}k11\\k1j\\k1i\\k1k\end{bmatrix} & \cdot\cdot & \begin{bmatrix}kj1\\kjj\\kji\\kjk\end{bmatrix} & \cdot\cdot & \begin{bmatrix}ki1\\kij\\kii\\kik\end{bmatrix} & \cdot\cdot & \begin{bmatrix}kk1\\kkj\\kki\\kkk\end{bmatrix} \end{bmatrix}.$$

Proposition II.2.2: The diagonal matrix $D = \text{diag}(a_1, a_2, \ldots, a_k)$, acts as follows:

$$
\mathcal{T}_D([M]) = \begin{bmatrix}
\begin{bmatrix} 11\cdot 111 \\ 11^2\cdot\overline{22}\cdot 112 \\ \vdots \\ 11^2\cdot\overline{kk}\cdot 11k \end{bmatrix}
\begin{bmatrix} 22\cdot 121 \\ 11\cdot 122 \\ \vdots \\ 11\cdot 22\cdot\overline{kk}\ 12k \end{bmatrix}
\cdots
\begin{bmatrix} kk\cdot 1k1 \\ 11\cdot\overline{22}\cdot kk\cdot 1k2 \\ \vdots \\ 11\cdot 1kk \end{bmatrix} \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\begin{bmatrix} kk\cdot k11 \\ kk\cdot\overline{22}\cdot 11\cdot k12 \\ \vdots \\ 11\cdot k1k \end{bmatrix}
\begin{bmatrix} 22\cdot kk\cdot\overline{11}\cdot k21 \\ kk\cdot k22 \\ \vdots \\ 22\cdot k2k \end{bmatrix}
\cdots
\begin{bmatrix} kk^2\cdot\overline{11}\cdot kk1 \\ kk^2\cdot\overline{22}\ kk2 \\ \vdots \\ kk\cdot kkk \end{bmatrix}
\end{bmatrix}.
$$

Proposition II.2.3: For $1 < i,j < k$, $i \neq j$, $\mathcal{T}_{E_{a_{ij}}}([M])$ is:

$$
\begin{array}{cc}
 & \\
\begin{bmatrix}
\begin{bmatrix} 111 \\ 11i - ij\cdot 11j \\ 11k \end{bmatrix} & \cdots &
\begin{bmatrix} ij\cdot 1i1 + 1j1 \\ ij\cdot 1ii - ij^2\cdot 1ij + 1ji - ij\cdot 1jj \\ ij\cdot 1ik + 1jk \end{bmatrix} & \cdots \\
\vdots & & \vdots & \\
\begin{bmatrix} ij\cdot i11 + j11 \\ ij\cdot i1i - ij^2\cdot i1j + j1i - ij\cdot j1j \\ ij\cdot i1k + j1k \end{bmatrix} & \cdot &
\begin{bmatrix} ij^2\cdot ii1 + ij\cdot ji1 + ij\cdot ij1 + jj1 \\ \begin{bmatrix} ij^2\cdot iii - ij^3\cdot iij + ij\cdot jii - ij^2\cdot jij + \\ ij\cdot iij - ij^2\cdot ijj + \quad jji - ij\cdot jjj \end{bmatrix} \\ ij^2\cdot iik + ij\cdot jik + ij\cdot ijk + jjk \end{bmatrix} & \\
\vdots & & \vdots & \\
\begin{bmatrix} k11 \\ k1i - ij\cdot k1j \\ k1k \end{bmatrix} & \cdots &
\begin{bmatrix} ij\cdot ki1 + kj1 \\ ij\cdot kii - ij^2\cdot kij + kji - ij\cdot kjj \\ ij\cdot kik + kjk \end{bmatrix} & \cdots
\end{bmatrix} \\
\text{column } 1 \qquad\qquad\qquad\qquad \text{column } j &
\end{array}
$$

Space prevents writing every row and column, but the pattern is clear: All rows of vectors except row i look like row 1 or k. Similarly, all columns of vectors except column j look like column 1 .

Note: For i = 1 and/or j = k, the effect is not quite the

same; nonetheless, the computation is straightforward, and will be made when necessary. These three have been shown for illustration.

3. Restriction to the ring case.

In this section, terminology defined in Section II.1 will be used in a new context; in fact, the two contexts are very closely related.

We will look at a group action on the set $\mathfrak{M}$ of multiplication tables on R.

Definition II.3.1: Let $A \in \mathrm{Tran}\,(\alpha_i)_{i=1}^{k}$, $\mathcal{B} = \{e_i\}$ in natural order. Let M be the multiplication table of μ with respect to $\mathcal{B}$. Define $A(\cdot):\mathfrak{M}\longrightarrow\mathfrak{M}$ such that

$$A(M) = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & A^T \end{array}\right] \cdot \left[\begin{array}{c|ccc} \mu & e_1 & \cdots & e_k \\ \hline e_1 & e_{11} & \cdots & e_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ e_k & e_{k1} & \cdots & e_{kk} \end{array}\right] \cdot \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & A \end{array}\right];$$

In shorthand, $A(M) = A^T{*}M{*}A$.

Observe A(M) is the multiplication table of μ on R^+ with respect to $\mathcal{B}' = \{\sum_{i=1}^{k} a_{i1}e_i, \ldots, \sum_{i=1}^{k} a_{ik}e_i\}$. The following properties hold:

(1) If M is an associative or commutative table, so is A(M).

(2) If $A,B \in \mathrm{Tran}\,(\alpha_i)_{i=1}^{k}$, then

$(AB)(M) \triangleq (AB)^T{*}M{*}(AB) = (B^TA^T){*}M{*}AB = B^T{*}A(M){*}B = B(A(M))$.

Thus A($\cdot$) is multiplicative, and hence is a group action on the set of multiplication tables.

(3) M_1 and M_2 represent the <u>same</u> multiplication μ (with respect to different bases) if and only if there exists $A \in \mathrm{Tran}\,(\alpha_i)^k_{i=1}$ such that $A(M_1) = M_2$.

There is a mapping analagous to A($\cdot$), except operating on cubes representing rings:

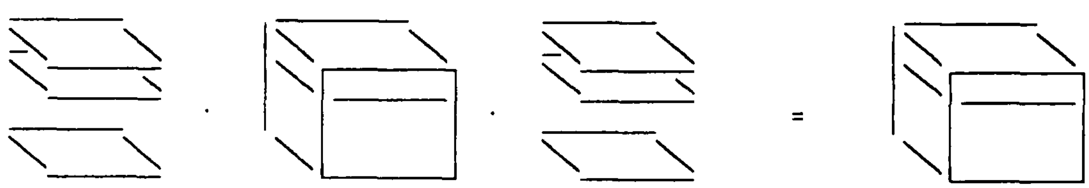
Definition II.3.2: Given $A \in \mathrm{Tran}\,(\alpha_i)^k_{i=1}$, M a multiplication table of μ on R^+ with respect to a basis $\mathcal{B}$, [M] its coordinate cube. Then A([M]) is defined to be:

$$\begin{bmatrix} A^T & 0 & 0 & 0 \\ 0 & A^T & 0 & 0 \\ 0 & & \ddots & 0 \\ 0 & 0 & 0 & A^T \end{bmatrix} \cdot \begin{bmatrix} [m^1_{ij}] & 0 & 0 & 0 \\ 0 & [m^2_{ij}] & & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & & [m^k_{ij}] \end{bmatrix} \cdot \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & & A \end{bmatrix}$$

In shorthand, this is

$(I_k \otimes A^T)\cdot \mathrm{diag}\,([M]^{\ell}_{**})\cdot(I_k \otimes A)$, $\ell = 1,\ldots,k$, where $\otimes$ is the Kronecker product of two matrices.

This operation can be visualized as follows:



$I \otimes A^T \quad \cdot \quad [M]_{\mathcal{B}} \quad \cdot \quad I \otimes A \quad = \quad A([M]_{\mathcal{B}})$

Figure II.1. Representation of A([M]) on the cube [M].

Proposition II.3.1: Given A in Tran $(\alpha_\iota)_{\iota=1}^{k}$, M a multiplication table of μ with respect to a basis $\mathcal{B}$, [M] its coordinate cube; then A([M]) is the coordinate cube for the multiplication table A(M) of μ but whose coordinates are with respect to $\mathcal{B}$.

Proof: For simplicity, the augmentations of A and A^T, and, in turn, the basis rows and columns of M can be dispensed with. Thus, by abuse of notation,

$$A(M) \triangleq \begin{bmatrix} a_{11} & \cdots & a_{k1} \\ \vdots & \ddots & \vdots \\ a_{1k} & \cdots & a_{kk} \end{bmatrix} \cdot \begin{bmatrix} e_{11} & \cdots & e_{1k} \\ \vdots & \ddots & \vdots \\ e_{k1} & \cdots & e_{kk} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix}$$

$$= \begin{bmatrix} \sum a_{\iota 1} e_{\iota 1} & \cdots & \sum a_{\iota 1} e_{\iota k} \\ \vdots & \ddots & \vdots \\ \sum a_{\iota k} e_{\iota 1} & \cdots & \sum a_{\iota k} e_{\iota k} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_j a_{j1} \sum_\iota a_{\iota 1} e_{\iota j} & \cdots & \sum_j a_{jk} \sum_\iota a_{\iota 1} e_{\iota j} \\ & \ddots & \\ \sum_j a_{j1} \sum_\iota a_{\iota k} e_{\iota j} & \cdots & \sum_j a_{jk} \sum_\iota a_{\iota k} e_{\iota j} \end{bmatrix}_{k \times k}.$$

Representing $e_{\iota j}$ by its coordinates $[m^u_{\iota j}]$, u=1,..,k, A(M) corresponds to the cube

$$[T]_{\mathcal{B}} = \begin{bmatrix} \sum_j a_{j1} \sum_\iota a_{\iota 1} [m^u_{\iota j}] & \cdots & \sum_j a_{jk} \sum_\iota a_{\iota 1} [m^u_{\iota j}] \\ \vdots & \ddots & \vdots \\ \sum_j a_{j1} \sum_\iota a_{\iota k} [m^u_{\iota j}] & \cdots & \sum_j a_{jk} \sum_\iota a_{\iota k} [m^u_{\iota j}] \end{bmatrix}_{k \times k \times k}, \quad \text{where } u=1,..,k$$

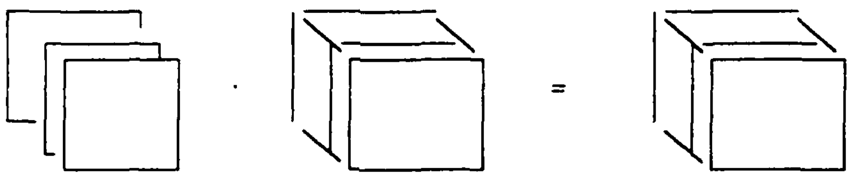
Note that $[T]^n_{**}$ is precisely

$$[T]^{n}_{**} = \begin{bmatrix} \sum_j a_{j1} \sum_i a_{i1} \cdot m^{n}_{ij} & \cdots & \sum_j a_{jk} \sum_i a_{i1} \cdot m^{n}_{ij} \\ \vdots & \ddots & \vdots \\ \sum_j a_{j1} \sum_i a_{ik} \cdot m^{n}_{ij} & \cdots & \sum_j a_{jk} \sum_i a_{ik} \cdot m^{n}_{ij} \end{bmatrix}_{k \times k}, \text{ which is}$$

$A^{T} \cdot [M]^{n}_{**} \cdot A$. Thus $[T]_{\mathcal{B}} = A([M]_{\mathcal{B}})$, which proves the Proposition. ■

In order to make use of Theorem I.3, there must be one more operation on the cube $A([M]_{\mathcal{B}})$ so that it represents $A(M)$ in the coordinate system $\mathcal{B}' = \{ \sum_{i=1} a_{i1} e_i, \ldots, \sum_{i=1} a_{ik} e_i \}$; namely, to multiply each coordinate stack by A^{-1}--specifically, one must find $A^{-1} \cdot \begin{bmatrix} m^{1}_{ij} \\ \vdots \\ m^{k}_{ij} \end{bmatrix}$ for every i,j=1,..,k.

This can be streamlined by computing $A^{-1} \cdot [A([M]_{\mathcal{B}})]^{*}_{i*}$ or $A^{-1} \cdot [A([M]_{\mathcal{B}})]^{*}_{*j}$ for each i or j=1,..,k. To illustrate:



$I \otimes A^{-1} \quad \cdot \quad A([M]_{\mathcal{B}}) \quad = \quad [A([M]_{\mathcal{B}})]_{\mathcal{B}'}$

Figure II.2. Picture of a change of basis.

This figure corresponds to $A^{-1} \cdot [A([M]_{\mathcal{B}})]^{*}_{i*}$. For $A^{-1} \cdot [A([M]_{\mathcal{B}})]^{*}_{*j}$, the slices would be left-to-right rather than back-to-front. The cube on the right corresponds to the multiplication table $A(M)$ with respect to the basis $\mathcal{B}'$.

The following is the main result of this section. The significance of this theorem is that the one-line formula defined by $\mathcal{T}_A([M])$ is a simple means of computing cubes representing rings which are equivalent (isomorphic) to the ring represented by $[M]_B$. Further, it ties together the abstract group action and the multiplication table of a ring. Finally, if R^+ is a free $\mathbb{Z}/p^d$ - module, it shows Mult $R^+ \cong \mathfrak{T}$; if R^+ is the direct sum of free modules, Mult R^+ is isomorphically imbedded in $\mathfrak{T}$.

Theorem II.1: Given R^+ with basis $\mathcal{B} = \{e_i\}_{i=1}^{k}$, $A \in$ Tran $(m_i)_{i=1}^{k}$, M_B and $[M]_B$ the multiplication table and cube for (R^+,μ) with respect to $\mathcal{B}$.Then

$\mathcal{T}_A([M]_B) = [A([M]_B)]_{B'}$, where $\mathcal{B}' = \{\sum_{i=1} a_{i1}e_i, \ldots, \sum_{i=1} a_{ik}e_i\}$.

Proof: First, evaluate $A([M]_B)$, which from Definition II.3.2 is:

$$\begin{bmatrix} A^T & 0 & 0 & 0 \\ 0 & A^T & 0 & 0 \\ 0 & & \ddots & 0 \\ 0 & 0 & 0 & A^T \end{bmatrix} \cdot \begin{bmatrix} [m^1_{ij}] & 0 & 0 & 0 \\ 0 & [m^2_{ij}] & & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & & [m^k_{ij}] \end{bmatrix} \cdot \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & & A \end{bmatrix}$$

Recall that this is the horizontal "slicing" operation depicted in Figure II.1. Thus for slice u of the cube $A([M]_B)$, we get

$$[A([M]_B)]^u_{ij} = \sum_{r=1} \sum_{s=1} a_{ri} a_{sj} m^u_{rs}. \tag{II.3.1}$$

This, as was said before, is the representation of the multiplication table A(M), but with coordinates in terms of the basis $\mathscr{B}$. To transform these coordinates to those of $\mathscr{B}'$, $[A([M]_B)]_{B'}$ must be computed. Thus we get

$$[[A([M]_B)]_{B'}]^u_{ij} = \sum_{r=1} \sum_{s=1} \sum_{t=1} a_{ri} a_{sj} \bar{a}_{ut} m^t_{rs}$$

which is how $[\mathcal{T}_A([M])]^u_{ij}$ is defined. This proves the theorem.■

This leads immediately to the following

Theorem II.2: Given an abelian group R^+ with basis $\mathscr{B} = \{e_i\}_{i=1}^k$, μ and ν two multiplications on R^+, [M] and [N] the respective coordinate cubes with respect to $\mathscr{B}$. Denote (R^+,μ) by R_1 and (R^+,ν) by R_2.

Then $R_1 \cong R_2$ if and only if there exists a matrix $A \in \mathrm{Tran}\,(m_i)_{i=1}^k$ such that $\mathcal{T}_A([M]) = [N]$.

Proof: By Theorem I.3, we know $R_1 \cong R_2$ if and only if there exists $A \in \mathrm{Tran}\,(m_i)_{i=1}^k$ such that $[M]_B = [N]_{B'}$. So, if $R_1 \cong R_2$, we get $[N]_B = A([M]_B)$, and Theorem II.1 above brings us the rest.□

The converse is straightforward, again by Theorem II.1 and then applying Theorem I.3.■

Remark 2.3: The development of the group action in the more general setting has an interesting application in the area of algebraic cryptography. Due to the peripheral nature of this subject to ring theory, only a brief discussion will be given.

One can think of a cube as a numeric representation of an alphanumeric message. The transition matrix would then serve as an encoding/decoding matrix, and the image $\mathcal{T}_A([M])$ would be the enciphered/deciphered text. In the usual cryptography problem, one is given [M] and $\mathcal{T}_A([M])$ and is asked to find A. The matrix representation $\mathcal{L}(A)$ suggests that there is no linear method of subdividing the cube in such a way that A can be easily found. In fact, the solvability of the quadratic identities, to be introduced in Chapter III, and the solvability of the cryptography problem are equivalent. In a future paper, the author hopes to present a cryptographic application of the operator $\mathcal{T}_A([M])$.

III. IDEMPOTENTS AND THE STRUCTURE OF FINITE RINGS

1. The Quadratric Identities.

Theorem II.2 can be restated in terms of matrix equations not involving A^{-1}. These equations are obtained by expressing the equation

$$\mathcal{T}_A([M]_B) = [A([M]_B)]_{B'} \overset{\Delta}{=} [N]_{B'} \qquad \text{(III.1.1)}$$

in terms of the basis $\mathcal{B}$ rather than $\mathcal{B}'$.

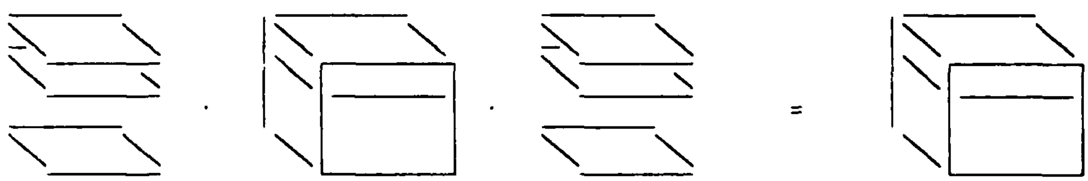
Theorem III.1: $R_1 \cong R_2$ if and only if the following system of equations are satisfied for some $A \in \text{Tran}\ (\alpha_\iota)_{\iota=1}^{k}$:

1) $[A^T[M]^{\ell}_{**}A]_{\iota *} = [A[N]^{*}_{\iota *}]_{\ell *}$ $\quad i,\ell = 1,..,k.$ (III.1.2)

2) $[A^T\cdot([M]^{\ell}_{**})^T\cdot A]_{j*} = [A[N]^{*}_{*j}]_{\ell *}$ $\quad j,\ell = 1,..,k.$ (III.1.3)

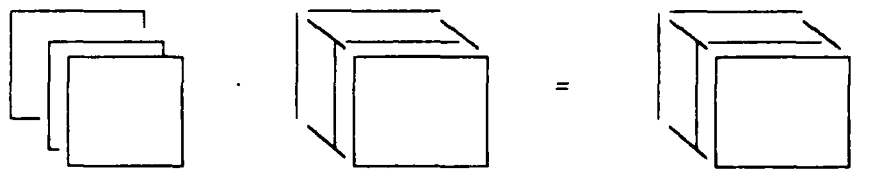
NOTE: These formulas will be referred to as the <u>quadratic identities.</u>

Proof: These equations are derived with the help of Figure II.1, which is reproduced here:



$I\otimes A^T \quad \cdot \quad [M]_B \quad \cdot \quad I\otimes A \quad = \quad A([M]_B)$

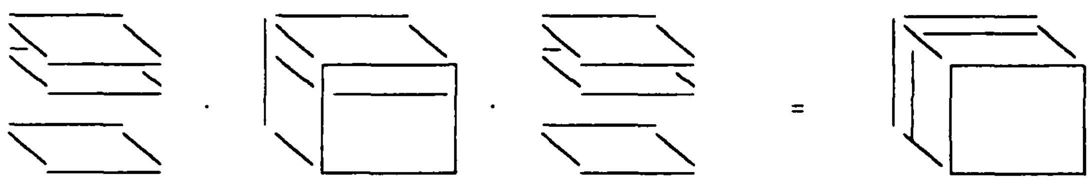
Instead of next performing the basis change operation to produce $[A([M]_B)]_{B'} = [N]_{B'}$, alter $[N]_{B'}$ instead by multiplying by $I\otimes A$ as pictured below:



$$I \otimes A \cdot [N]_{\mathcal{B}'} = A([M]_{\mathcal{B}})$$

Figure III.1. Modified form of Basis Change Step.

This gives us the equality $A([M]_{\mathcal{B}}) = [N]_{\mathcal{B}}$ instead. Notice that both cubes are expressed in terms without reference to A^{-1}. Looking again at the pictures below, (the cubes are with respect to $\mathcal{B}$)



$$I \otimes A^{T} \cdot [M] \cdot I \otimes A = A \cdot [N]^{*}_{i*}$$

Figure III.2. Figures II.1 and III.1 combined.

the ℓ-th horizontal slice of each cube is given by Equation III.1.2). One obtains Equation 2) similarly, substituting $[N]^{*}_{*j}$ for $[N]^{*}_{i*}$.∎

These quadratic identities are the basic computational tool that will be used and modified in the rest of this paper. The term "quadratic identity" refers to the fact that each of III.1.2 and III.1.3 corresponds to a system of equations of degree two in the variables a_{ij}. The system is given by:

$$\left.\begin{aligned} &\sum_{u=1} \sum_{v=1} a_{vi} a_{uc} m^{\ell}_{vu} - \sum_{u=1} a_{u\ell} \cdot n^{u}_{ic} = 0, \quad i,\ell,c=1,..,k \\ &\sum_{u=1} \sum_{v=1} a_{vi} a_{uc} m^{\ell}_{uv} - \sum_{u=1} a_{u\ell} \cdot n^{u}_{ci} = 0, \quad i,\ell,c=1,..,k \end{aligned}\right\} \quad \text{(III.1.4)}$$

where c means the c-th column of the vectors in equations III.1.2 and III.1.3 in Theorem III.1. It is understood that all arithmetic is performed over $\mathbb{Z}/p^{\alpha_1}$.

An important property of a ring--namely, possession of an identity--can be tested as an application of Theorem III.1. In fact, in order to test for an identity, the quadratic identities reduce to a system of linear equations in k variables. This result is new, and is a direct result of the tools developed in this paper. The next theorem is a description of the method. Its significance is that it provides a mechanism for identifying a particular transition matrix which reveals **1** as a basis element.

Theorem III.2: R is a ring with identity if and only if the following systems of equations are satisfied for some $A \in \mathrm{Tran}\,(\alpha_i)_{i=1}^{k}$:

1) $[a_{11}\ a_{21}\ \cdots\ a_{k1}]\cdot[M]^{\ell}_{**} = I_{\ell *}$ $\qquad \ell = 1,..,k.$

2) $[a_{11}\ a_{21}\ \cdots\ a_{k1}]\cdot([M]^{\ell}_{**})^{T} = I_{\ell *}$ $\qquad \ell = 1,..,k.$

Proof: $\Rightarrow$: If **1** $\in$ R, then as noted in Remark I.1.3, $\langle \mathbf{1} \rangle$ can be made into a direct summand of R^{+} and hence **1**

would be a basis element. Proceeding as in Theorem I.1, a new basis $\mathscr{B}'$ for R^+ is obtained, and thus there exists an $A \in \mathrm{Tran}\,(\alpha_\iota)_{\iota=1}^{k}$ such that A would be the appropriate basis change matrix. By Theorem III.1, R is a ring with **1** if and only if the cubes $[M]_B$ and $\mathcal{T}_A([M]_B)$ have the property that

$$[A^T \cdot [M]^L_{**} \cdot A]_{*1} = [A^T \cdot [M]^L_{**} \cdot A]_{1*} = A_{L*} \quad L = 1,..,k. \qquad (III.1.5)$$

The cube below describes this equation :

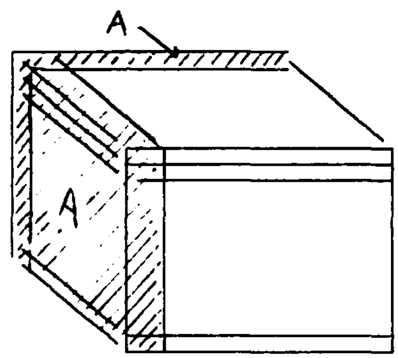


Figure III.3. "Picture" of Equation III.1.5.
Note that **A** is on the left and rear faces of the cube.

What this means is that row L of A is precisely the first row or column of the L-th slice-matrix $A^T \cdot [M]^L_{**} \cdot A$. Therefore, R is a ring with **1** if and only if the following string of equalities hold (L=1,..,k):

$$[a_{L1} \cdots a_{Lk}] = [1\ 0 \cdots 0] \cdot [A^T \cdot [M]^L_{**} \cdot A] \qquad (III.1.6)$$

$$[a_{L1} \cdots a_{Lk}] \cdot A^{-1} = [1\ 0 \cdots 0] \cdot [A^T \cdot [M]^L_{**} \cdot A] \cdot A^{-1} \qquad (III.1.7)$$

$$[0\ 0 \cdots \underset{\substack{\uparrow \\ \text{L-th entry}}}{1} \cdots 0] = [1\ 0 \cdots 0] \cdot A^T \cdot [M]^L_{**} \qquad (III.1.8)$$

$$[0\ 0\ \cdots\ \underset{\substack{\uparrow\\ L\text{-th entry}}}{1}\ \cdots\ 0] = [a_{11}\ a_{21} \cdots a_{k1}] \cdot [M]^{L}_{**}. \qquad \text{(III.1.9)}$$

This last equation implies that A can be determined if and only if the k distinct systems in Equation III.1.9 have a common suitable solution $[a_{11}\ a_{21} \cdots a_{k1}]$. Should such a solution exist, any member of Tran $(m_\iota)^{k}_{\iota=1}$ whose first column is this solution will produce

$[A(M)]^{*}_{*1} = [A(M)]^{*}_{1*} = I_k.$

If no such common suitable solution exists, then R does not have a 1. □

⇐: The converse argument is basically the reverse of ⇒ and is easily followed.■

EXAMPLE III.1.1:

Let $R = \mathbb{Z}_3[x]/_{(x^3+2x+1)}$, the field of 27 elements. Let $\mathcal{B} = \{1+x, 2+x^2, 1+2x+2x^2\}$. Then

$$M_{\mathcal{B}} = \begin{array}{c|ccc} & 1+x & 2+x^2 & 1+2x+2x^2 \\ \hline 1+x & 1+2x+x^2 & 1+x & 2+2x+x^2 \\ 2+x^2 & 1+x^2 & 1+2x+2x^2 & x+x^2 \\ 1+2x+2x^2 & 2+2x+2x^2 & x+x^2 & 2+2x \end{array}$$

and

$$[M]_B = \begin{bmatrix} \begin{bmatrix}2\\1\\0\end{bmatrix} & \begin{bmatrix}2\\0\\2\end{bmatrix} & \begin{bmatrix}0\\2\\1\end{bmatrix} \\ \begin{bmatrix}2\\0\\2\end{bmatrix} & \begin{bmatrix}0\\0\\1\end{bmatrix} & \begin{bmatrix}1\\1\\0\end{bmatrix} \\ \begin{bmatrix}0\\2\\1\end{bmatrix} & \begin{bmatrix}1\\1\\0\end{bmatrix} & \begin{bmatrix}2\\0\\0\end{bmatrix} \end{bmatrix}$$

From this cube, it is not clear that R has a **1**. By the Theorem, R does, provided a solution to the systems of equations..

$$[1\ 0\ 0] = [a_{11}\ a_{21}\ a_{31}] \cdot \begin{bmatrix} 2 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$[0\ 1\ 0] = [a_{11}\ a_{21}\ a_{31}] \cdot \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} \qquad \text{(III.1.10)}$$

$$[0\ 0\ 1] = [a_{11}\ a_{21}\ a_{31}] \cdot \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

exists.

Since the cube is commutative, the other three equations are redundant and are not shown. That this has a unique common solution of $[1\ 1\ 1]^T$ is easy to compute, and hence for $A = \begin{bmatrix} 1 & * & * \\ 1 & * & * \\ 1 & * & * \end{bmatrix}$ where A is nonsingular mod p, $[\mathcal{T}_A([M])]$ is as desired. In fact, for $A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix}$, $[\mathcal{T}_A([M])]$ is the cube for R with respect to the basis $\mathcal{B}' = \{1, x, x^2\}$.□

Remark 3.1:

1. The Quadratic Identities represent an always-consistent system of quadratic equations in the variables a_{ij}, $i,j = 1,\ldots,k$. This consistency does not guarantee that the matrix or matrices identified as solutions are in $\mathrm{Tran}\,(m_i)_{i=1}^{k}$. However, this is easily checked. If none of the matrices which solve the Quadratic Identities are in $\mathrm{Tran}\,(d_i)_{i=1}^{k}$, then the rings are not isomorphic.

2. The quadratic identities, by themselves, would not be a very effective computational tool because of the intractibility of multivariate quadratic equations. What is interesting, however, is that, except for operations over $\mathbb{Z}/_{2}d$, these identities can be treated as a linear system of $2k^3$ homogeneous equations in $(k^4+k^2)/2 + k^2$ linearly independent variables. That is, the monomials $a_{vi}a_{uc}$ are all linearly independent of a_{ij}, except over $\mathbb{Z}/_{2^d}$. $\mathbb{Z}/_{2^d}$ is restricted because the equation $2^{d-1}a_{ij} + 2^{d-1}(a_{ij})^2 \equiv 0 \bmod 2$ holds for $a_{ij} \in \mathbb{Z}/_{2^d}$, and so a_{ij} and $(a_{ij})^2$ are not linearly independent. A look at the numbers show that rank 2 and 3 rings can be attacked by this method.

3. For every a_{ij} that can be determined, either explicitly

or in terms of other a_{uv}'s, k^2 of the original quadratic variables are reduced to either constants or other monomials. This effectively collapses the system to a significantly smaller problem. The next example illustrates this point.

Example III.1.2: Suppose two cubes [M] and [N] are known to represent rings R_1 and R_2, respectively, and that both rings possess a 1, and are of rank 3, characteristic 3. Then both cubes can be changed so that $R_1' = \langle 1\rangle \oplus \langle e_2\rangle \oplus \langle e_3\rangle$ and $R_2' = \langle 1\rangle \oplus \langle f_2\rangle \oplus \langle f_3\rangle$. Then $R_1' \cong R_2'$ if and only if there exists $A \in \mathrm{Tran}\ (\alpha_\iota)_{\iota=1}^{k}$ of the form $\begin{bmatrix} 1 * * \\ 0 * * \\ 0 * * \end{bmatrix}$ which solves the "linear" system of 54 equations in 54 unknowns. Knowing $a_{11} = 1$, $a_{21} = 0$, and $a_{31} = 0$, then we know $(a_{\iota 1})^2$, $a_{\iota 1}a_{j1}$ to be constants and $a_{\iota 1}a_{uv}$ to be either a_{uv} or 0, depending on $a_{\iota 1}$. Thus, the system of quadratic equations now have 27 fewer variables to be contended with than the original 54.

2. The Trace Identities.

The well-known trace function for matrices is useful in the present discussion, chiefly because it is a linear function. The next theorem is a weaker result than Theorem III.1; however, when they are used together, they form an effective computational tool.

Theorem III.3: If $R_1 \cong R_2$, then the following systems of equations are consistent: (the a_{ij} are from the transition matrix A)

$$\sum_{v=1}^{k} \mathrm{Tr}[M]^*_{v*} a_{vi} = \mathrm{Tr}[N]^*_{i*} \quad , \; i = 1,..,k. \qquad \text{(III.2.1)}$$

$$\sum_{v=1}^{k} \mathrm{Tr}[M]^*_{*v} a_{vj} = \mathrm{Tr}[N]^*_{*j} \quad , \; j = 1,..,k. \qquad \text{(III.2.2)}$$

NOTE: These last equations will be referred to as the <u>Trace Identities</u>.

Pf: Assume $R_1 \cong R_2$. Then by Theorem III.1, the quadratic identities hold, which are (for review):

$$[A^T \cdot [M]^{\ell}_{**} \cdot A]_{i*} = [A \cdot [N]^*_{i*}]_{\ell *} \qquad i,\ell=1,..,k, \qquad \text{(III.1.2)}$$

$$[A^T \cdot ([M]^{\ell}_{**})^T \cdot A]_{j*} = [A \cdot [N]^*_{*j}]_{\ell *} \qquad j,\ell=1,..,k. \qquad \text{(III.1.3)}$$

Working with (III.1.2), and ranging over ℓ, we have

$$\begin{bmatrix} [A^T \cdot [M]^1_{**} \cdot A]_{i*} \\ \vdots \\ [A^T \cdot [M]^k_{**} \cdot A]_{i*} \end{bmatrix} = A \cdot [N]^*_{i*} \qquad i =1,..,k. \qquad \text{(III.2.3)}$$

$$\begin{bmatrix} [a_{1i} \; .. \; a_{ki}]\cdot[M]^{1}_{**}\cdot A \\ \vdots \\ [a_{1i} \; .. \; a_{ki}]\cdot[M]^{k}_{**}\cdot A \end{bmatrix} = A\cdot[N]^{*}_{i*} \qquad i = 1,..,k, \quad (III.2.4)$$

$$\begin{bmatrix} [a_{1i} \; .. \; a_{ki}]\cdot[M]^{1}_{**} \\ \vdots \\ [a_{1i} \; .. \; a_{ki}]\cdot[M]^{k}_{**} \end{bmatrix}\cdot A = A\cdot[N]^{*}_{i*} \qquad i = 1,..,k. \quad (III.2.5)$$

Multiplying both sides on the right by A^{-1}, and using the multiplicativity of the trace function, we get

$$\mathrm{Tr}\begin{bmatrix} [a_{1i} \; .. \; a_{ki}]\cdot[M]^{1}_{**} \\ \vdots \\ [a_{1i} \; .. \; a_{ki}]\cdot[M]^{k}_{**} \end{bmatrix} = \mathrm{Tr}\;[N]^{*}_{i*} \qquad i = 1,..,k. \quad (III.2.6)$$

The desired Trace Identities follow.■

Equations III.2.1 and III.2.2,

$$\left\{ \begin{array}{lll} 1) \sum_{v=1}^{k} \mathrm{Tr}[M]^{*}_{v*} a_{vi} = \mathrm{Tr}[N]^{*}_{i*} & , & i = 1,..,k \\ 2) \sum_{v=1}^{k} \mathrm{Tr}[M]^{*}_{*v} a_{vj} = \mathrm{Tr}[N]^{*}_{*j} & , & j = 1,..,k \end{array} \right\}$$

can be expressed as a system of equations:

$$\begin{bmatrix} a_{11} & a_{21} & .. & a_{k1} \\ a_{12} & a_{22} & .. & a_{k2} \\ \vdots & & \ddots & \vdots \\ a_{1k} & a_{2k} & .. & a_{kk} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Tr}[M]^{*}_{1*} \\ \mathrm{Tr}[M]^{*}_{2*} \\ \vdots \\ \mathrm{Tr}[M]^{*}_{k*} \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}[N]^{*}_{1*} \\ \mathrm{Tr}[N]^{*}_{2*} \\ \vdots \\ \mathrm{Tr}[N]^{*}_{*j} \end{bmatrix}.$$

using Equation III.2.1, and a similar system arises using equation III.2.2. These equations can be "turned on their end" where the matrix A^{T} becomes a k^{2}-long column vector,

and the Tr[M] vector becomes a block diagonal matrix; i.e.,

$$\left[\begin{array}{c|c|c} \mathrm{Tr}[M]^{*}_{1*} \,..\, \mathrm{Tr}[M]^{*}_{1*} & & 0 \\ \hline & \cdot & \\ & \cdot & \\ \hline 0 & & \mathrm{Tr}[M]^{*}_{1*} \;..\; \mathrm{Tr}[M]^{*}_{1*} \end{array}\right] \cdot \begin{bmatrix} a_{11} \\ \vdots \\ \vdots \\ a_{kk} \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}[N]^{*}_{1*} \\ \\ \\ \mathrm{Tr}[N]^{*}_{k*} \end{bmatrix}.$$

There is a corresponding system for the second equation as well; obviously we would solve them simultaneously, producing a system of 2k equations in k^2 unknowns, and the block-diagonal format of the system insures at most k^2-k free variables in the solution space, should it be consistent.

That the converse of Theorem III.2 doesn't hold is shown by the following example.

Example III.2.1: Let R^+_1 be the direct sum $\langle e_1\rangle \oplus \langle e_2\rangle \oplus \langle e_3\rangle$ such that R_1 is the trivial ring of 27 elements, and let R^+_2 be the direct sum $\langle 1\rangle \oplus \langle e_2\rangle \oplus \langle e_3\rangle$, where

$$e_2 \cdot e_3 = e_3 \cdot e_2 = 0 = e_2^{\,2} = e_3^{\,2}.$$

Both are characteristic 3 rings. Both have 0 traces on all slices, but clearly the rings are nonisomorphic.□

The Trace Identities' main defect is that a solution to them may or may not satisfy the more conclusive quadratic identities. On the other hand, any solution not admitted by the Trace Identities cannot be a solution to the

quadratic identities. Further, the Trace Identities may be an inconsistent system of linear equations; in this case, we know that the rings under consideration will not be isomorphic.

Because of the structure of the linear system induced by the Trace Identities, there are upper and lower bounds to the dimension of the solution space; namely,

$$k^2-2k \leq \text{dim sol'n space} \leq k^2-k,$$

provided the system is consistent.

3. A test for idempotents in R.

Thus far, associativity, commutativity, and possession of an identity have been tested by direct computation. In this section several new results relating to decomposing a p-ring into its irreducible components are obtained by identifying central and one-sided idempotents.

Definition III.3.1: An _idempotent_ is an element $e \in R$ such that $e^2 = e$. It is _central_ provided it commutes with every element of the ring. It is _nontrivial_ if it is not 0_R or 1_R.

Definition III.3.2: A ring R is _nil_ provided for all $x \in R$, $x^{k(x)} = 0_R$ for some nonnegative integer k(x). It is nilpotent provided $R^k = 0$ for some positive integer k.

It is known that if R is finite then nil and nilpotent rings coincide.

Proposition III.3.1: A finite ring R is nilpotent if and only if the only idempotent in the ring is 0_R.

Proof: ⇒: A nilpotent ring cannot contain any nonzero idempotent, because $e^2 = e \neq 0_R \Rightarrow e^k \neq 0_R$ for all nonnegative integers k.□

⇐: Suppose R has no nonzero idempotents and that R is not nilpotent. Then there exists an element x whose powers

never vanish. Since R is finite, x has the property that $x^s = x^t$ for some $s > t$. This implies that there exists a $y = x^r$ such that $y^2 = y$ for some positive integer r.

Pf of assertion: In case $s \geq 2t$, we have $x^{s+k} = x^{t+k}$ for all $k \geq 1$. Then at some point $s+k = 2(t+k)$. Let $r = t+k$.

In case $s < 2t$, note that $x^{t^2} = x^{tt} = x^{ts} = x^{ss} = x^{s^2}$. Since $s > t$, this process can be repeated until $s^{2k} > 2t^{2k}$ for some k, and then revert back to the first case.□

This assertion establishes the contradiction. Hence there is no such nonvanishing element. Thus R is nilpotent.■

The next theorem, adapted from [Ja], helps to determine the reducibility of nonnilpotent rings.

Theorem III.4: If a finite nonnilpotent ring R possesses a nontrivial central idempotent **e**, then $R \cong \mathbf{e}\cdot R \oplus (1-\mathbf{e})\cdot R$, where $(1-\mathbf{e})\cdot R = \{r - \mathbf{e}r \mid r \in R\}$ and $\mathbf{e}\cdot R$ are two-sided ideals of R. Further,

1. $\mathbf{e}\cdot R$ is a ring with identity **e**;
2. $(1-\mathbf{e})\cdot R$ is a ring with identity if and only if R is a ring with identity;
3. $\mathbf{e}\cdot(1-\mathbf{e})\cdot R = 0 = (1-\mathbf{e})\cdot R\cdot \mathbf{e}$; i.e., **e** annihilates $(1-\mathbf{e})\cdot R$.

Proof: A straightforward calculation.■

A noncentral idempotent will reduce a ring into a direct sum of right ideals $\mathbf{e}\cdot R \oplus (1-\mathbf{e})\cdot R$, but not into two-sided ideals. While $\mathbf{e}\cdot R$ is a subring of R, $(1-\mathbf{e})\cdot R$ is not. Nonetheless, noncentral idempotents have some properties that are useful in a cube setting, which will be shown shortly.

From here on, rings will be analyzed with respect to the presence or absence of the various idempotents. The next group of theorems provide a means of determining whether or not a particular cube possesses a central or noncentral idempotent.

Proposition III.3.2: A finite p-ring R possesses a nontrivial central idempotent if and only if

1) there exists a sequence of $E_1,..,E_k \in \text{Tran}\ (\alpha_i)_{i=1}^{k}$ such that $\mathcal{T}_{E_1 \cdots E_k}([M]) = [N]$ has the property that $[N]^*_{ii}$ is the i-th column vector of **I** for some i=1,..,k, and $[N]^*_{i*} = [N]^*_{*i}$;

2) $[N]^*_{i*}$ is not the identity.

Proposition III.3.3: A finite p-ring R possesses a noncentral idempotent if and only if there exists a

sequence of $E_1,..,E_k \in \mathrm{Tran}\ (\alpha_i)_{i=1}^{k}$ such that

$\mathcal{T}_{E_1 \cdots E_k}([M]) = [N]$ has the property that for some $i=1,..,k$,

$[N]^*_{i*}$ is the i-th column of **I**, and $[N]^*_{i*} \neq [N]^*_{*i}$.

These are both clear from the definition of a nontrivial central/noncentral idempotent.

Proposition III.3.4: If **e** is a central idempotent for R, then **e** can be made into a basis element for R.

Proof: Because of Theorem III.3, we have **e**·R is a ring with identity. If **e**·R = R, then **e** is the identity of the ring. If **e**·R ≠ R, then (1-**e**)·R is nontrivial. Construct bases for each of these subrings. Because **e** is the identity of **e**·R, it can be made into a basis element of the ideal **e**·R (see Remark I.1.1), which in turn makes it a basis element of R, since R is factored into these two ideals.■

Proposition III.3.5: Let **e** be a noncentral idempotent of R. Then R is a direct sum of right ideals **e**·R ⊕ (1-**e**)·R, where (1-**e**)·R is as before. Further,

1. **e** is a left identity of **e**·R and an annihilator of (1-**e**)·R from the left.

2. **e** can be made into a basis element for R.

The proof follows the same lines as Theorem III.3 and Proposition III.3.5, noting that a left (or right) identity has maximal additive order in a ring and thus can be made into a basis element for **e**·R and R·**e**, respectively.■

The consequences of Propositions III.3.4 and Theorem III.4 is that the cubes of such rings have a certain form, shown next:

Proposition III.3.6: If R is a ring with central idempotent, then depending on the rank of **e**·R, any cube [M] representing R is equivalent to

$$[N] = \begin{array}{c}\begin{bmatrix} \begin{bmatrix}1\\0\\ \vdots\\0\end{bmatrix} & \begin{bmatrix}0\\1\\ \vdots\\0\end{bmatrix} & & \begin{bmatrix}0\\ \vdots\\1\\ \vdots\\0\end{bmatrix} & \begin{bmatrix}0\\ \vdots\\0\\ \vdots\\0\end{bmatrix} & & \begin{bmatrix}0\\ \vdots\\0\\ \vdots\\0\end{bmatrix} \\ \begin{bmatrix}0\\1\\ \vdots\\0\end{bmatrix} & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & \cdots & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & \cdots & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} \\ \begin{bmatrix}0\\ \vdots\\0\\ \vdots\\0\end{bmatrix} & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} & & \begin{bmatrix}*\\ *\\ \vdots\\ *\end{bmatrix} \end{bmatrix} \\ \uparrow \\ \text{col } i \end{array}$$

where i is the rank of the subring **e**·R.

Proof: It follows from Theorem III.3.■

The next result is one of the more important ones in this paper. Previous propositions on idempotents relied on

existence of a sequence of transition matrices. The next theorem gives a method of telling if certain rings have an idempotent by merely inspecting the cube.

Theorem III.5: Let R be a ring of type ${}_p(d_1,..,d_k)$, $d_1,..,d_k$ a <u>strictly</u> decreasing sequence, $\mathcal{B}$ its basis, and $[M]_{\mathcal{B}}$ its cube. Then R possesses a nonzero idempotent if and only if at least one of the vectors $[M]^*_{ii}$, $i = 1,..,k$ has the property that m^i_{ii} is a unit.

Proof: $\Leftarrow$: Suppose m^i_{ii} is a unit. Then it is claimed that $(\mathbf{e}_i)^k \neq 0_R$ for all $k \geq 1$.

Proof of assertion: Observe that $(\mathbf{e}_i)^2 = m^1_{ii}\mathbf{e}_1+..+m^i_{ii}\mathbf{e}_i+..+m^k_{ii}\mathbf{e}_k$. Because R is a ring, we have $p^{d_j-d_i}$ divides m^j_{ii} for all $j < i$. Because m^i_{ii} is a unit, $o((\mathbf{e}_i)^2) = o(\mathbf{e}_i)$. Observe that $(\mathbf{e}_i)^3 = m^1_{ii}\mathbf{e}_1\mathbf{e}_i+..+m^i_{ii}(\mathbf{e}_i)^2+..+m^k_{ii}\mathbf{e}_k\mathbf{e}_i$. Because $o(\mathbf{e}_{ji}) \leq o(\mathbf{e}_i)$ for all $j \leq i$, we thus have $o((\mathbf{e}_i)^3) = o(\mathbf{e}_i)$. Proceeding inductively, the assertion is proved.□

From here, following the argument in Proposition III.3.1, there exists an integer r such that $(\mathbf{e}_i)^{2r} = (\mathbf{e}_i)^r$. Thus $(\mathbf{e}_i)^r$ is an idempotent for R, and the first half of the

proposition is proved.□

⇒: Suppose R possesses an idempotent e and that, for contradiction, [M] has the property that m^i_{ii} is divisible by p for all i=1,..,k. It will be shown is that <u>every</u> basis in natural order for R will have this property; that is, under any change of basis which preserves natural order, $[\mathcal{T}_A([M])]^i_{ii}$ is divisible by p for all i=1,..,k. This will imply that there is no e such that $e^2 = e$ because of Propositions III.3.2 and III.3.3. Since every transition matrix is a product of elementary matrices described in Chapter II.2, it will suffice to analyze them.

It is easy to show that diag$\{a_{11},..,a_{kk}\}$ and $E_{a_{ij}}$, i < j, preserve this divisibility by p. So consider $E_{a_{ij}}$, i > j, and compute $[\mathcal{T}_A([M])]^t_{tt}$ for t=1,..,k. By tabulation, we have:

a. $[\mathcal{T}_A([M])]^t_{tt} = m^t_{tt}$ when $t \neq i,j$.

b. $[\mathcal{T}_A([M])^j_{jj} = 1\cdot jjj + ij\cdot jij + ij\cdot ijj + ij^2\cdot iij$, using the notation of Chapter II.2. Without loss of generality, we can assume that a_{ij} is a unit. By hypothesis, p divides jjj. By Corollary I.3.1a, p divides jij. By Definition I.2.2, p divides iij. Because $o(e_{ij}) \le o(e_i)$ (Theorem I.2), we have p divides ijj. Thus p divides $[\mathcal{T}_A([M])^j_{jj}$.

c. $[\mathcal{T}_A([M])]^i_{ii} = 1\cdot iii - ij\cdot iij$. Again, by hypothesis, p divides iii, and by Definition I.2.2, p divides iij. Thus

p divides $[\mathcal{T}_{\blacktriangle}([M])]^{i}_{ii}$.
Since all cases are exhausted, the contradiction is established and the Theorem is proven.■

The case of rings where $d_i = d_{i+1}$ for some i poses some difficulties. Nonetheless, some results can be obtained, whose proofs follow lines similar to the previous theorem.

Proposition III.3.7: Let R be a ring of type $_p(d_1,..,d_k)$ with $\mathcal{B}$ a basis in natural order. Suppose $d_i = d_{i+1}$ for some i=1,..,k-1. Then the following holds.

1. If m^{i}_{ii} is the sole unit of $[M]^{*}_{ii}$ and/or $m^{i+1}_{i+1,i+1}$ is the sole unit of $[M]^{*}_{i+1,i+1}$, or if for some d_j which is unique, m^{j}_{jj} is a unit, then R possesses an idempotent.
2. If p divides all of $[M]^{*}_{ii}$ and all of $[M]^{*}_{i+1,i+1}$, and if for all unique d_j, p divides m^{j}_{jj}, then R is nilpotent.■

By observing the proof of the converse of Theorem III.5, a method for actually finding a basis change matrix which exhibits the idempotent e as a basis element can be derived. This method can also be applied when the conditions of Proposition III.3.7 are satisfied. The following is a discussion of the method:

Note that in case m^{i}_{ii} is a unit for some i (d_i is unique), then we have $o(e_i^{k}) = o(e_i)$ for all $k \geq 1$. Thus, as was

noted earlier, there exists s and t, s > t, such that $(e_i)^s = (e_i)^t$. Before proceeding further note that $(e_i)^3 = (e_i)^2 \cdot e_i$ (R is associative) which implies $(e_i)^3 = m_{ii}^1 e_{1i} + m_{ii}^2 e_{2i} + .. + m_{ii}^k e_{ki}$, which corresponds to

$$m_{ii}^1 \cdot \begin{bmatrix} m_{1i}^1 \\ \vdots \\ m_{1i}^k \end{bmatrix} + m_{ii}^2 \cdot \begin{bmatrix} m_{2i}^1 \\ \vdots \\ m_{2i}^k \end{bmatrix} + .. + m_{ii}^k \cdot \begin{bmatrix} m_{ki}^1 \\ \vdots \\ m_{ki}^k \end{bmatrix},$$ which can be

re-expressed as

$$\begin{bmatrix} m_{1i}^1 & m_{2i}^1 & & m_{ki}^1 \\ \vdots & \cdot & \cdots & \vdots \\ m_{1i}^k & m_{2i}^k & & m_{ki}^k \end{bmatrix} \cdot \begin{bmatrix} m_{ii}^1 \\ \\ m_{ii}^k \end{bmatrix},$$ which is $[M]_{*i}^* \cdot [M]_{ii}^*$. Denote,

for brevity, the vector representation of $(e_i)^k$ by v_k, and $[M]_{*i}^*$ by M. Then we have

$v_3 = Mv_2$, $v_4 = Mv_3 = M^2 v_2$, etc. The point of the proof of Theorem III.5 is that once it is shown that $(e_i)^s = (e_i)^t$, one can find an r such that $(e_i)^{2r} = (e_i)^r$. Hence, we also have $v_r = v_{2r} = Mv_{2r-1} = M^r v_r$; that is, v_r is an "eigenvector" of M^r and 1 is its corresponding "eigenvalue". (The terms are in quotes because this is not necessarily a vector space setting, and so the terms aren't strictly defined. The concept is analogous, however.) More importantly, Because α_i is unique, and because $o(e_i^{2r}) = o(e_i^r) = o(e_i)$, e_i^r can be made a basis element replacing e_i, and further, it is an <u>idempotent</u>. Thus, if $\mathcal{B} = \{e_1, .., e_i, .., e_k\}$ is a basis for R, then

$\mathcal{B}' = \{e_1, .., (e_i)^r, .., e_k\}$ is also a basis, and if

$$[e_i^r]_B = \begin{bmatrix} n_{ii}^1 \\ \vdots \\ n_{ii}^k \end{bmatrix}, \text{ the matrix } A = \begin{bmatrix} 1 & 0 & n_{ii}^1 & 0 \\ 0 & 1 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & n_{ii}^k & 1 \end{bmatrix} \text{ is the}$$

transition matrix from $\mathcal{B}$ to $\mathcal{B}'$.

Note that with repsect to $\mathcal{B}'$, e_i^{2r} has coordinates $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$,

which is as desired.■

The next Proposition enables one to actually create a basis for $e \cdot R$ and $(1-e) \cdot R$, once e is identified, provided R satisfies the condition of Theorem III.5. Thus, one can systematically produce cube forms similar to that of Proposition III.3.6 without having to rely on the quadratic identities.

Proposition III.3.8: Let e be a nonzero idempotent for R, R a ring satisfying the conditions of Theorem III.5. Let $\mathcal{B}$ be a basis for R. Then either $e \cdot e_i$ can replace e_i as a basis element for R, or $e_i - e \cdot e_i$ can.

Proof: It is assumed that e has been made into a basis element, and that only the remaining basis elements are

being considered. For clarity, $\mathcal{E} = \{e_1,..,e,..,e_k\}$ is in natural order. The proof rests on analyzing the additive order of $e \cdot e_i$, $i = 1,..,k$.

Since d_i is unique, then $o(e \cdot e_i) = o(e_i)$ implies a straight substitution can be made. If $o(e \cdot e_i) < o(e_i)$, then $o(e_i - e \cdot e_i) = o(e_i)$, and the substitution $e_i - e \cdot e_i$ for e_i can be made.■

Remark III.3.1: There is again some difficulty when d_i is not unique. The author was not able to fully resolve this case, though some partial results, not worthy of mention here, were obtained. The difficulty lies when $o(e \cdot e_{i+h}) = o(e_{i+h})$, where $d_i = d_{i+h}$. The issue of linear independence must be examined, and this is very difficult to do.

Nonetheless, a large number of ring types can be computationally analyzed for number and type of idempotents by the methods developed in this section. This fills a gap previously existing in the literature; namely, how to systematically identify and display idempotents in an arbitrary ring.

There is one more case that can be proven, but in order to express it, a different form must be used.

Proposition III.3.9: Let R be a ring of type ${}_p(d,d)$. Let [M] be of the form $\begin{bmatrix} \begin{bmatrix} m^1_{11} \\ m^2_{11} \end{bmatrix} & \begin{bmatrix} m^1_{12} \\ m^2_{12} \end{bmatrix} \\ \begin{bmatrix} m^1_{21} \\ m^2_{21} \end{bmatrix} & \begin{bmatrix} m^1_{22} \\ m^2_{22} \end{bmatrix} \end{bmatrix}$ where m^j_{ii} are units for $i,j = 1,2$. Let $A = \begin{bmatrix} 1 & (m^2_{11})^{-2} m^1_{11} \\ 0 & (m^2_{11})^{-1} \end{bmatrix}$, so that $\mathcal{T}_A([M])$ is $\begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} n^1_{12} \\ n^2_{12} \end{bmatrix} \\ \begin{bmatrix} n^1_{21} \\ n^2_{21} \end{bmatrix} & \begin{bmatrix} n^1_{22} \\ n^2_{22} \end{bmatrix} \end{bmatrix}$. Then:

1. If either n^1_{22} or n^2_{22} are units, <u>but not both</u>, then R posseses an idempotent.
2. If both are divisible by p, then R is nilpotent.
3. If both n^1_{22} and n^2_{22} are units, no conclusion can be reached by mere inspection. Another method, such as the quadratic identities, must be used.

Proof: 1 and 2 follow from the previous proposition. As for 3, the outcome depends on p and the values of the rest of the cube. It should be noted that all elements of the cube $\mathcal{T}_A([M])$ are units except the zero depicted.■

IV--A DISCUSSION OF COMPUTER ALGORITHMS TO TEST FOR RING PROPERTIES AND FOR ISOMORPHISM BETWEEN TWO RINGS

This chapter contains a discussion of the algorithms described in Chapters I-III, programmed in Turbo PASCAL™ [Bo], and listed in the Appendix. Examples of their execution are included as well.

Because of the action of the elementary matrix S_{ij} on a cube (Proposition II.3.2), all bases will be assumed to be in natural order.

1. The program which computes $\mathcal{T}_A([M])$.

The program SLICER.PAS computes the mapping $\mathcal{T}_A([M])$ given the following inputs:

1. The type of R^+;
2. The cube [M], by rows;
3. The transition matrix A.

The output would be as follows:

1. The transition matrix and its inverse mod p^{d_1},
2. The cube [M] and its image $\mathcal{T}_A([M])$.

The rank of the ring and the prime number are considered to be program constants, and can be changed only by program modification. At present, only keyboard entry of input data is provided.

The heart of the program is the computation of A^{-1} mod p^{d_1}, should it exist, and Equation II.1.1,

$[\mathcal{T}_A([M])]^u_{ij} = \sum_{r=1} \sum_{s=1} \sum_{t=1} a_{ri} a_{sj} \bar{a}_{ut} m^t_{rs}$. The standard matrix inversion method must be modified to accomodate $\mathbb{Z}/p^{d_1}$; that is, the pivot for any column must be a unit mod p, not just any nonzero element. The algorithm reflects this difference. As for Equation II.1.1, it is a six-index loop involving the inputs A and [M], and the computed A^{-1}. All computations are made in integer mod p^{d_u} arithmetic, except for A^{-1}, which is performed in mod p^{d_1} arithmetic.

Example IV.1.1. Let the rank of R^+ be $_5(3,2,2,1)$. Let

$$A = \begin{bmatrix} 111 & 40 & 105 & 75 \\ 44 & 103 & 2 & 50 \\ 35 & 73 & 55 & 25 \\ 78 & 8 & 77 & 1 \end{bmatrix}.$$

This is a transition matrix, for it satisfies the requirements of Proposition I.2.2.

$$\text{Let } [M] = \begin{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{matrix}$$

Then SLICER, which computes $\mathcal{T}_A([M])$, produces

$$\mathcal{T}_A([M]) = \begin{matrix} \begin{bmatrix} 76 \\ 4 \\ 16 \\ 0 \end{bmatrix} & \begin{bmatrix} 70 \\ 10 \\ 4 \\ 2 \end{bmatrix} & \begin{bmatrix} 120 \\ 6 \\ 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 70 \\ 10 \\ 4 \\ 2 \end{bmatrix} & \begin{bmatrix} 85 \\ 8 \\ 3 \\ 3 \end{bmatrix} & \begin{bmatrix} 65 \\ 7 \\ 17 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 120 \\ 6 \\ 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 65 \\ 7 \\ 17 \\ 1 \end{bmatrix} & \begin{bmatrix} 110 \\ 18 \\ 3 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{matrix} . \square$$

2. The program which checks the basic properties of a cube [M].

BASPROPS.PAS tests if a given cube represents a multiplication for a group R^+ of type ${}_p(d_1,..,d_k)$. It then checks for associativity and commutativity. The values k and p are again treated as program constants, and the input values are the type of R^+ and the cube [M]. The entries m^t_{rs} are reduced mod p^{d_t} if necessary, and the residue is given on the screen if such action is taken.

The first procedure checks that [M] represents a multiplication. A three-index loop checking k^3-k^2 elements (m^t_{rs}, $r < s$) for divisibility by $p^{d_r-d_s}$ is made. The procedure first checks if $d_1 = d_k$, in order to

eliminate unnecesary computation.

The next procedure computes $[M]^*_{i*} \cdot [M]^*_{*j} - [M]^*_{*j} \cdot [M]^*_{i*}$ for each $i,j = 1,..,k$. As soon as one of the entries of this matrix is nonzero, the procedure returns a "nonassociative" response. Thus, the program is optimized for failure.

Example IV.2.1. The author wrote a modified version of this program to find all rings with 1 of characteristic 3. Of the 3^{12} different multiplications on the basis $\{1,e_2,e_3\}$, only 802 of the cubes proved to be associative as well. The program, written in FORTRAN and run on NCSUMATH, took 13 minutes of processing time.□

The final test checks each horizontal slice of [M] for symmetry by computing $m^t_{rs} - m^t_{sr}$ for $r \neq s$. Again, a nonzero result immediately returns a "noncommutative" response.

3. The program which test for the existence of a 1 in R. IDENTITY.PAS checks a cube for the existence of an identity, providing sufficient information to draw the correct conclusion. If necessary, one can construct a suitable transition matrix so that if $\mathcal{B} = \{e_1,..,e_k\}$, then $\mathcal{B}' = \{1,f_2,..,f_k\}$. The input is as in BASPROPS.PAS.

This program builds a $2k^z \times (k+1)$ linear system and then reduces it to upper triangular form. The results are printed so that column 1 of the transition matrix A can be obtained, in accordance with Corollary III.1 and Example III.1.1.

In the case R^+ is a vector space over $\mathbb{Z}/_p$, the procedure is the well-known Gaussian elimination. However, just as in IV.1, in case R^+ is not of type ${}_p(1,..,1)$, the elimination method must be modified to first seek a pivot that is a unit; should one not exist, then a pivot of minimum p-valuation is sought; that is, a nonzero element with the lowest power of p in that column.

Example IV.3.1: Suppose R^+ is of type ${}_5(4,2)$. Let

$[M] = \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}75\\1\end{bmatrix} \end{bmatrix}$. It is clearly a ring with **1**. The test for identity gives rise to the following system of equations:

$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 75 & 0 \\ 0 & 5 & 0 \\ 5 & 5 & 5 \end{array}\right]$. What makes this reduction different is that 5 divides 75 and is not a unit. Thus, when finding the pivot for column 2, one must first attempt to find a unit. The

transition matrix is $\begin{bmatrix} 1 & * \\ 0 & * \end{bmatrix}$, as expected. For another example, see Example III.1.2.□

4. A discussion of the quadratic identities.

QUADID.PAS is the implementation of the quadratic and trace identities to compare two rings for isomorphism. A scaled-down version of this program, IDEMPOT.PAS can be used to check for nontrivial idempotents. The basic inputs are the type of R^+, two cubes, and responses to questions provided through interactive screens. The outpu consists of a printout of the quadratic system of equations after each set of questions is answered. For review, the quadratic and trace identities are, respectively:

$$[A^T[M]^{\ell}_{**}A]_{i*} = [A[N]^{*}_{i*}]_{\ell*} \qquad i,\ell = 1,..,k. \quad \text{(III.1.2)}$$

$$[A^T\cdot([M]^{\ell}_{**})^T\cdot A]_{j*} = [A[N]^{*}_{*j}]_{\ell*} \qquad j,\ell = 1,..,k. \quad \text{(III.1.3)}$$

and

$$\sum_{v=1}^{k} Tr[M]^{*}_{v*}a_{vi} = Tr[N]^{*}_{i*} \qquad , i = 1,..,k. \qquad \text{(III.2.1)}$$

$$\sum_{v=1}^{k} Tr[M]^{*}_{*v}a_{vj} = Tr[N]^{*}_{*j} \qquad , j = 1,..,k. \qquad \text{(III.2.2)}$$

Observe that Equations III.1.2 and III.1.3 are equations of vectors; thus each component of these vectors must equal as well. That is, for each $i,\ell = 1,..,k$,

$$
\begin{aligned}
[A^T[M]^{\ell}_{**}A]_{i1} &= [A[N]^{*}_{i*}]_{\ell 1},\\
[A^T[M]^{\ell}_{**}A]_{i2} &= [A[N]^{*}_{i*}]_{\ell 2},\\
&\vdots\\
[A^T[M]^{\ell}_{**}A]_{ik} &= [A[N]^{*}_{i*}]_{\ell k}
\end{aligned}
$$

Letting c be the variable corresponding to the columns of the vectors, the equations III.1.4 arise, which for review, are:

$$
\left.
\begin{aligned}
&\sum_{u=1} \sum_{v=1} a_{vi}a_{uc}m^{\ell}_{vu} - \sum_{u=1} a_{u\ell}\cdot n^{u}_{ic} = 0, \quad i,\ell,c=1,..,k\\
&\sum_{u=1} \sum_{v=1} a_{vi}a_{uc}m^{\ell}_{uv} - \sum_{u=1} a_{u\ell}\cdot n^{u}_{ci} = 0, \quad i,\ell,c=1,..,k
\end{aligned}
\right\} \qquad \text{(III.1.4)}
$$

The question which must be answered first is, How big is this system? There are at most $2k^3$ distinct equations. Viewing a_{11}, a_{12}, $a_{11}a_{12}$, .., $a_{kk}a_{kk}$ as variables, we see they are linearly independent, provided the characteristic of the ring is odd (see Remark III.1.2). Thus, there are $k^2 + \frac{k^2(k^2+1)}{2}$ distinct variables. A 2 × 2 example will illustrate:

Example IV.4.1: The general form of the quadratic identities involving two commutative cubes [M] and [N] representing rings of rank 2 is:

$$
\begin{array}{cccc}
\begin{array}{cccccccccc}
\multicolumn{4}{c}{a_{11}} & \multicolumn{3}{c}{a_{12}} & \multicolumn{2}{c}{a_{21}} & a_{22} \\
a_{11} & a_{12} & a_{21} & a_{22} & a_{12} & a_{21} & a_{22} & a_{21} & a_{22} & a_{22}
\end{array}
& & \begin{array}{cccc} a_{11} & a_{12} & a_{21} & a_{22} \end{array} & \text{const} \\
\left[\begin{array}{c|c|c|c|c|c|c|c|c|c}
m^1_{11} & 0 & m^1_{12}+m^1_{21} & 0 & 0 & 0 & 0 & m^1_{22} & 0 & 0 \\
0 & m^1_{11} & 0 & m^1_{12} & 0 & m^1_{21} & 0 & 0 & m^1_{22} & 0 \\
m^2_{11} & 0 & m^2_{12}+m^2_{21} & 0 & 0 & 0 & 0 & m^2_{22} & 0 & 0 \\
0 & m^2_{11} & 0 & m^2_{12} & 0 & m^2_{21} & 0 & 0 & m^2_{22} & 0 \\
0 & m^1_{11} & 0 & m^1_{21} & 0 & m^1_{12} & 0 & 0 & m^1_{22} & 0 \\
0 & 0 & 0 & 0 & m^1_{11} & 0 & m^1_{12}+m^1_{21} & 0 & 0 & m^1_{22} \\
0 & m^2_{11} & 0 & m^2_{21} & 0 & m^2_{12} & 0 & 0 & m^2_{22} & 0 \\
0 & 0 & 0 & 0 & m^2_{11} & 0 & m^2_{12}+m^2_{21} & 0 & 0 & m^2_{22}
\end{array}\right]
& - &
\left[\begin{array}{c|c|c|c}
n & n & 0 & 0 \\
n & n & 0 & 0 \\
0 & 0 & n & n \\
0 & 0 & n & n \\
n & n & 0 & 0 \\
n & n & 0 & 0 \\
0 & 0 & n & n \\
0 & 0 & n & n
\end{array}\right]
& = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right]
\end{array}
$$

Since the rings (cubes) are commutative, the transpose equations are redundant and hence are omitted.

The trace identities III.2.1 and III.2.2 have the following form:

$$
\left[\begin{array}{c|c|c|c}
s_{1*} & s_{2*} & 0 & 0 \\
s_{*1} & s_{*2} & 0 & 0 \\
\hline
0 & 0 & s_{1*} & s_{2*} \\
0 & 0 & s_{*1} & s_{*2}
\end{array}\right]
=
\left[\begin{array}{c}
t_{1*} \\
t_{*1} \\
\hline
t_{2*} \\
t_{*2}
\end{array}\right] . \square
$$

(s_{i*} and s_{*j} represent Tr $[M]^*_{i*}$ and Tr $[M]^*_{*j}$, respectively; t corresponds to $[N]$)

The algorithm for QUADID.PAS proceeds as follows:

1. Given the two cubes, the appropriate traces are computed and then stored in the proper array location. In the trace identities, the "inverse" array described in Chapter III is also stored.

2. Both trace identities systems are reduced to upper triangular form and then analyzed for consistency. If they are consistent, then the values for a_{ij} obtained by the first trace system are employed in the quadratic system.

3. The quadratic system is reduced to upper triangular form. The option of replacing a_{ij} by a constant c or by $d_1 - d_2 \cdot a_{uv}$ is offered. If $a_{ij} = c$, then each entry of the column corresponding to $a_{ij}a_{uv}$ is multiplied by c and moved to the column corresponding to a_{uv}. The entries of column a_{ij} are then multiplied by c and moved "across the equal sign".

If $a_{ij} = d_1 - d_2 \cdot a_{uv}$, then each entry of the column corresponding to $a_{ij}a_{vx}$ is multiplied by d_2 and then moved to the column corresponding to $a_{uv}a_{vx}$. Then the array is modified by d_1 in the same manner as in the first case.

The next example illustrates the use of QUADID.PAS, showing each key step and printout along the way.

Example IV.4.2: Consider two rings of rank 2, of type $_3(1,1)$, which are isomorphic, whose cubes are

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \end{bmatrix} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 2 \\ 0 \end{bmatrix} \end{bmatrix}. \quad \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \text{ is a transition}$$

matrix which transforms the first into the second. These cubes are entered into the program. The following

printout results:

Trace Formula Matrices   Sys2 For Information Only

```
 11  21  12  22  Col

  0   1   0   0   1          1   1   0   0   0
  0   1   0   0   1          1   1   0   0   0
  0   0   0   1   1          0   0   1   1   1
  0   0   0   1   1          0   0   1   1   1
```

Reduced Trace Formula Matrices

```
  0   1   0   0   1          1   1   0   0   0
  0   0   0   1   1          0   0   1   1   1
  0   0   0   0   0          0   0   0   0   0
  0   0   0   0   0          0   0   0   0   0
```

Notice that the trace identities reveal that $a_{21} = 1$ and $a_{22} = 1$. Continuing with the program, and using Example IV.4.1 as a guide, the first quadratic system output is:

```
Raw Tableau entries.
 0 0 1 0 0 0 0 2 0 0 0 1 0 0 0
 0 0 0 2 0 2 0 0 2 0 2 2 0 0 0
 0 0 0 0 0 0 0 2 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 2 0 0 0 2 2 0
 0 0 0 2 0 2 0 0 2 0 2 2 0 0 0
 0 0 0 0 0 0 1 0 0 2 1 0 0 0 0
 0 0 0 0 0 0 0 0 2 0 0 0 2 2 0
 0 0 0 0 0 0 0 0 0 2 0 0 1 0 0
Run Number 1
Reduced Tableau entries.
 0 0 1 0 0 0 0 2 0 0 0 1 0 0 0
 0 0 0 1 0 1 0 0 1 0 1 1 0 0 0
 0 0 0 0 0 0 1 0 0 2 1 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0
 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0
 0 0 0 0 0 0 0 0 0 1 0 0 2 0 0
```

Now, using the information about a_{*1}, the system is reduced to the following: (zero rows are not printed)

Run Number 2
Reduced Tableau entries.

0	0	0	1	0	0	0	0	0	0	1	2	0	1	0
0	0	0	0	0	0	1	0	0	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

From here, we observe that $a_{11} = 1-a_{12}$. A check by SLICER.PAS of $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, and $\begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}$ verifies that indeed these matrices serve as transition matrices for the equivalent cubes described above.□

Observation: One can use QUADID.PAS to find the stabilizer of a cube [M] by entering it twice in the program when called for. Finding the stabilizer by exhaustive means requires checking all transition matrices; an upper bound for the number of transition matrices is the number of nonsingular matrices over $\mathbb{Z}/p^{d_1}$, which, according to Roby [Ro] is given by

$$(p^{d_1})^{k^2} \cdot \prod_{i=1}^{k} \left(1 - \frac{1}{p^i}\right).$$

When R is a free $\mathbb{Z}/p^d$ - module, this number is exact.

Example IV.4.3: In the case of a ring with 1 of type $_3(1,1,1)$, a FORTRAN program run on NCSUMATH takes 72

seconds to check all 11232 nonsingular matrices for membership in the stabilizer of an arbitrary cube.□

V. SOME RESULTS FOR RINGS OF RANK 1 AND 2.

1. Rings of rank 1--Complete Classification.

This first result is known ([W2], [KP]).

Proposition V.1.1: Let R be a finite p-ring of rank 1, $|R| = p^n$. Then there are exactly n+1 mutually nonisomorphic rings, one with identity, and the rest nilpotent. The representative forms are:

$$[M] = [0],$$

$$[M] = [p^i],\ i=0,..,n-1.$$

Proof: By Theorem II.2, two rings are isomorphic if and only if there exists $A \in \mathrm{Tran}\ (\alpha_i)_{i=1}^{k}$ such that $\mathcal{T}_A([M]) = [N]$. These cubes are 1x1x1 in size; i.e, they are scalars. Thus Theorem II.2 means that $a \cdot m = n$ for some a such that $(a,p) = 1$. The cube of any finite p-ring R of rank one is of the form $[p^i \cdot u]$, where $(u,p) = 1$, or [0] in the case that $R^2 = 0$. choose $a = u^{-1}$, and the Proposition follows.■

2. Rings of cardinality p^2--Complete Classification.

The next few results gives all rings of size p^2. Raghavendran [Ra] found them by purely algebraic means. They are also mentioned in [W2], though not all rings are explicitly identified.

Proposition V.2.1: Let R be a ring of size p^2, rank one. Then there are three possible outcomes: [M] = [0], [1], or [p].

Proof: This is an immediate consequence of the previous Proposition.■

Proposition V.2.2: Let R be a ring of type $_p(1,1)$ without a nonzero idempotent (thus nilpotent). Then R is either the trivial ring or its cube is equivalent to

$$[M] = \begin{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix}.$$

Proof: Clearly, $R^2 = 0$ is a nilpotent ring. So suppose R is nilpotent and nontrivial. Then $e_i \cdot e_j = m_1e_1 + m_2e_2 \neq 0$ for some i,j. We first want to show that either e_1^2 or e_2^2 is nonzero. Assume both are zero. Without loss of generality, assume i=1 and j=2. Then $e_1 \cdot e_2 \neq 0 \Rightarrow$

$0 = e_1 \cdot e_1 \cdot e_2 = m_1e_1^2 + m_2e_1 \cdot e_2 = m_1m_2e_1 + m_2^2e_2$. Since m_2 cannot be zero, $(m_2,p) = 1 \Rightarrow m_2^2 \neq 0$, a contradiction. ($m_2$ can't be zero because if it is, then $e_1 \cdot e_2 = m_1e_1 \Rightarrow 0 = e_1 \cdot e_2^2 = m_1^2e_1 \neq 0$, also a contradiction.) Therefore, one of e_1^2 or e_2^2 is nonzero. Without loss of generality, assume $(e_1)^2$ is nonzero. Thus, we have the following "string" of cubes: (the reasons will be given below)

$$\begin{bmatrix} \begin{bmatrix} m^1_{11} \\ m^2_{11} \end{bmatrix} & \begin{bmatrix} m^1_{12} \\ m^2_{12} \end{bmatrix} \\ \begin{bmatrix} m^1_{21} \\ m^2_{21} \end{bmatrix} & \begin{bmatrix} m^1_{22} \\ m^2_{22} \end{bmatrix} \end{bmatrix} \overset{(1)}{\Rightarrow} \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} n^1_{12} \\ n^2_{12} \end{bmatrix} \\ \begin{bmatrix} n^1_{21} \\ n^2_{21} \end{bmatrix} & \begin{bmatrix} n^1_{22} \\ n^2_{22} \end{bmatrix} \end{bmatrix} \overset{(2)}{\Rightarrow} \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} a \\ b \end{bmatrix} \\ \begin{bmatrix} a \\ b \end{bmatrix} & \begin{bmatrix} ab \\ a+b^2 \end{bmatrix} \end{bmatrix}.$$

m^2_{11} is nonzero and thus a unit; otherwise e_1 would be an idempotent. Then (1) follows by computing $\mathcal{T}_A([M])$ with

$$A = \begin{bmatrix} 1 & m^1_{11}(m^2_{11})^{-2} \\ 0 & (m^2_{11})^{-1} \end{bmatrix}.$$

(2) follows from the associativity property of R, Proposition I.3.2. Because R is nontrivial and nilpotent, we have $R^3 = 0$, (powers of R form a strictly descending sequence of subrings terminating at 0). This forces $e_2{}^2 = e_1{}^4 = 0$ which implies $ab = a+b^2 = 0 \Rightarrow a = b = 0$, which completes the proof of the proposition.■

Proposition V.2.3: Let R be a ring of type ${}_p(1,1)$ with only noncentral idempotents. Then all such rings have cubes

equivalent to

$$[M] = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix} \text{ or } [M] = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}.$$

Proof: Since R is nonnilpotent and has no central idempotents, it is irreducible, then we have the following sequence of cubes:

$$\begin{bmatrix} \begin{bmatrix} m^1_{11} \\ m^2_{11} \end{bmatrix} & \begin{bmatrix} m^1_{12} \\ m^2_{12} \end{bmatrix} \\ \begin{bmatrix} m^1_{21} \\ m^2_{21} \end{bmatrix} & \begin{bmatrix} m^1_{22} \\ m^2_{22} \end{bmatrix} \end{bmatrix} \overset{(1)}{\Rightarrow} \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} n^1_{12} \\ n^2_{12} \end{bmatrix} \\ \begin{bmatrix} n^1_{21} \\ n^2_{21} \end{bmatrix} & \begin{bmatrix} n^1_{22} \\ n^2_{22} \end{bmatrix} \end{bmatrix} \overset{(2)}{\Rightarrow}$$

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} a \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ a \end{bmatrix} \end{bmatrix} \text{ or } \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} a \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ a \end{bmatrix} \end{bmatrix}.$$

(1) follows from Proposition III.3.4. Because R is associative, by Proposition I.3.2, n^2_{12} and n^2_{21} are 0 or 1 mod p and $n^1_{12}n^2_{12} = n^1_{21}n^2_{21} = 0$. e_1 is noncentral, so $e_2 \cdot e_1 \neq e_1 \cdot e_2$. Associativity then forces $n^1_{22} = 0$ and $n^2_{22} = n^1_{21}$ or n^1_{12} as appropriate. Let

$A = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 \\ -a & 1 \end{bmatrix}$, respectively. This forces $\mathcal{T}_A([M])$ to be one of the desired forms, and so the Proposition follows.■

Proposition V.2.4: Let R be a ring of type $_p(1,1)$ with at

least one central idempotent. Then the cube of R is equivalent to one of the following forms:

$$[M] = \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix} \text{ or } \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \end{bmatrix} \text{ or } \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix} \text{ or } \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} \end{bmatrix}.$$

Proof: Suppose that R has at least one nontrivial central idempotent. Then R is reducible by Theorem III.2 and thus is a direct sum of rings of rank one of size p, of which there are two types. This provides the first two listed possibilities. In the first case, R is a ring without **1**, and in the second, a ring with **1**. Now suppose R is a ring with **1** and no other central idempotent. Since rings of rank two with **1** are commutative (proof is obvious from the cube), then R is commutative, which means that there are no noncentral idempotents. Then [M] is equivalent to the following cube:

$$[M] = \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}a\\b\end{bmatrix} \end{bmatrix},$$ where $a \neq 0$ if $b \neq 0$ (else $\mathbf{e}_2$ is an idempotent). Then the fourth form is produced by $\mathcal{T}_A([M])$ where $A = \begin{bmatrix} a^{-1} & 0 \\ ba^{-2} & a^{-1} \end{bmatrix}$, and the third arises when both a and b are zero. This completes the proof of the Proposition.■

Thus, there are 11 isomorphism classes of rings of size p^2, for Propositions V.2.1 - V.2.4 exhaust all possible cases.

3. Rings of type $_p(\alpha_1,\alpha_2)$, $x^2 = 0$ for all x in R.

The above analysis is immensely more difficult when R is a rank of type $_p(\alpha_1,\alpha_2)$, where $\alpha_1 > 1$, as the next few sections will make clear. The results of this section are new and are similar to those found in **[KP]**.

Recall that by Proposition I.3.1, the general form of a cube representing a multiplication over a group of type $_p(\alpha_1,\alpha_2)$ is as follows:

$$[M] = \begin{bmatrix} \begin{bmatrix} m^1_{11} \\ m^2_{11} \end{bmatrix} & \begin{bmatrix} p^{\alpha_1-\alpha_2}c^1_{12} \\ m^2_{12} \end{bmatrix} \\ \begin{bmatrix} p^{\alpha_1-\alpha_2}c^1_{21} \\ m^2_{21} \end{bmatrix} & \begin{bmatrix} p^{\alpha_1-\alpha_2}c^1_{22} \\ m^2_{22} \end{bmatrix} \end{bmatrix}.$$ It is clear that we can restrict c^1_{ij} to $\mathbb{Z}/p^{\alpha_2}$ with no harm to the discussion. Unlike the $_p(1,1)$ case, a nontrivial nilpotent ring <u>can</u> have all its elements be square-zero. The next Proposition gives the cube form for those rings with this property.

Proposition V.3.1: Let R be of type $_p(\alpha_1,\alpha_2)$ with basis $\mathfrak{B}$ in natural order, and [M] its cube. Let $x^2 = 0\ \forall\ x \in R$. Then [M] is equivalent to a cube of the form

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \alpha p^{i_1} \\ \beta p^{j_1} \end{bmatrix} \\ \begin{bmatrix} \gamma p^{i_2} \\ \delta p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix} \qquad \begin{cases} \alpha,\beta = 0,1 \\ \gamma,\delta \in \mathbb{Z}/p \quad \text{when } i_1 - j_1 \geq i_2 - j_2 \\ d_1 - d_2 \leq i_1 < j_1 + d_1 - d_2 \leq d_1 \end{cases}$$

-or-

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \alpha p^{i_1} \\ \beta p^{j_1} \end{bmatrix} \\ \begin{bmatrix} \gamma p^{i_2} \\ \delta p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix} \qquad \begin{cases} \gamma,\delta = 0,1 \\ \alpha,\beta \in \mathbb{Z}/p \quad \text{when } i_1 - j_1 < i_2 - j_2 \\ d_1 - d_2 \leq i_2 < j_2 + d_1 - d_2 \leq d_1 \end{cases}$$

NOTE: The second form will be known as the <u>mirror</u> of the first. The two forms are not necessarily antiisomorphic.

Proof: Since $x^2 = 0$, we have every basis element being square-zero, so that [M] must be of the form (allowing for Corollary I.3.1)

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} a p^{i_1} \\ b p^{j_1} \end{bmatrix} \\ \begin{bmatrix} c p^{i_2} \\ d p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix} \quad \text{where} \quad \begin{cases} d_1 - d_2 \leq i_1, i_2 \leq d_1 \\ 0 \leq j_1, j_2 \leq d_2 \end{cases}$$

and $a,b,c,d \in \mathbb{Z}/p$. The proof now breaks down into several cases. Due to the length of each, they will be noted in italics.

Case 1a: a,b,c,d nonzero, $i_1 - i_2 \geq j_1 - j_2$.

Then we have the following sequence of cubes:

$$\begin{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}ap^{i_1}\\bp^{j_1}\end{bmatrix}\\ \begin{bmatrix}cp^{i_2}\\dp^{j_2}\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix}\end{bmatrix} \xrightarrow{(1)} \begin{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}p^{i_1}\\p^{j_1}\end{bmatrix}\\ \begin{bmatrix}\vartheta p^{i_2}\\\delta p^{j_2}\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix}\end{bmatrix} \xrightarrow{(2)} \begin{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\p^{j_1}\end{bmatrix}\\ \begin{bmatrix}\gamma p^{i_2}\\\delta p^{j_2}\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix}\end{bmatrix}.$$

(1) follows by letting $A = \begin{bmatrix}b^{-1} & 0\\0 & a^{-1}\end{bmatrix}$; then $\vartheta = a^{-1}c$ and

$\delta = b^{-1}d$. (2) follows by letting $B = \begin{bmatrix}1 & p^{i_1 - j_1}\\0 & 1\end{bmatrix}$, *provided*

$i_1 - i_2 \geq d_1 - d_2$. (This qualifies B as a transition matrix.)

γ is a unit whenever $i_1 - i_2 > j_1 - j_2$, and possibly 0 when

$i_1 - i_2 = j_1 - j_2$. *Finally*, when γ is a unit, for

$C = \begin{bmatrix}1 & 0\\0 & \gamma^{-1}\end{bmatrix}$, $\mathcal{T}_{A\cdot B\cdot C}([M])$ becomes $\begin{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\p^{j_1}\end{bmatrix}\\ \begin{bmatrix}p^{i_2}\\\delta p^{j_2}\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix}\end{bmatrix}$, for

which associativity implies $j_1 \geq \frac{d_2}{2}$.

Now suppose $i_1 - j_1 < d_1 - d_2$. In this case, matrix B would not be a transition matrix, and so (2) would <u>not</u> follow; hence, the only modification we can make is with the units a,b,c and d. Then (1) is the most basic form possible, and the restriction $d_1 - d_2 \leq i_1 < j_1 + d_1 - d_2 \leq d_1$ holds.□

Case 1b. $i_1 - j_1 < i_2 - j_2$. By the same methods as in case 1a, we get the "mirror" form. □

Case 2. $a = 0$, $b,c,d \neq 0$.

Then [M] is of the form $\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ bp^{\jmath_1} \end{bmatrix} \\ \begin{bmatrix} cp^{\imath_2} \\ dp^{\jmath_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}$, and for

$A = \begin{bmatrix} b^{-1} & 0 \\ 0 & c^{-1} \end{bmatrix}$, $\mathcal{T}_A([M])$ becomes $\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ p^{\jmath_1} \end{bmatrix} \\ \begin{bmatrix} p^{\imath_2} \\ \delta p^{\jmath_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}$, which is one

of the forms described in case 1a. $a = 0$ means $\imath_1$ is understood to be equal to d_1, and this form holds provided $d_1 - d_2 \leq \imath_2 < \jmath_2 + d_1 - d_2 \leq d_1$. If, however, $\imath_2 - \jmath_2 \geq d_1 - d_2$, then one can replace $p^{\imath_2}$ by 0, which is achieved by use

of the transition matrix $\begin{bmatrix} \delta^{-1} & 0 \\ (p^{\imath_2 - \jmath_2}) & 1 \end{bmatrix}$. □

Case 3. $c = 0$, $a,b,d \neq 0$.

This is the "Mirror" case to case 2. □

Case 4. $a = b = 0,\ c,d \neq 0.$

Then [M] is of the form $\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} cp^{i_2} \\ dp^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}$, and for $A = \begin{bmatrix} d^{-1} & 0 \\ 0 & c^{-1} \end{bmatrix}$,

$\mathcal{T}_A([M])$ becomes $\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} p^{i_2} \\ p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}$, which, as in the previous case,

depends on whether or not $d_1 - d_2 \leq i_2 < j_2 + d_1 - d_2 \leq d_1$.□

Case 5. $a,b \neq 0,\ c=d=0.$

This case is resolved as in case 4. When $d_1 > d_2$, the forms of cases 4 and 5 are antiisomorphic. When $d_1 = d_2$, they are equivalent.

Case 6. $a,c = 0,\ b,d \neq 0.$

Then [M] is of the form $\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ bp^{j_1} \end{bmatrix} \\ \begin{bmatrix} 0 \\ dp^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}$, which becomes

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ p^{j_1} \end{bmatrix} \\ \begin{bmatrix} 0 \\ \delta p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \ \delta \in \left(\mathbb{Z}/p\right)^* \text{ by the methods of earlier cases.}\square$$

Case 7. a,b≠0, c=d=0.

Thye general form, after similar modification, is seen to be

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} p^{i_1} \\ 0 \end{bmatrix} \\ \begin{bmatrix} \gamma p^{i_2} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \delta \in \left(\mathbb{Z}/p\right) \ .\square$$

The last cases, where all but one constant is zero, are easily seen to be equivalent to one of

$$\begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} p^{i_1} \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \quad \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ p^{j_1} \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \quad \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} p^{i_2} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \text{or} \quad \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ p^{j_2} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}.\blacksquare$$

In order to properly count the number of mutually nonisomorphic cases, one must know the value of d_1-2d_2 and the relationships between i_1-j_1, i_2-j_2, and d_1-d_2. As a result, a closed formula for counting the number of isomorphism classes is very difficult to obtain.

4. Rings of type $_p(d_1,d_2)$ with nontrivial central idempotent.

This classification can be quickly made, using Theorem III.2. Surprisingly, the author could find no reference to it in the literature.

Proposition V.4.1: Let R be a ring of type $_p(d_1,d_2)$, **e** a nontrivial central idempotent. Then [M] has one of the following two forms:

$$(1)\quad \begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\p^i\end{bmatrix} \end{bmatrix} \; 0\le i\le d_2 \qquad (2)\quad \begin{bmatrix} \begin{bmatrix}p^j\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \end{bmatrix} \; 0\le j\le d_1$$

In case $d_1 = d_2$, then the two forms are equivalent, with the transition matrix being $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$.

Proof: Since **e** is a nontrivial central idempotent, it is not O or 1. By Theorem III.2, **e** is the identity for the two-sided ideal **e**·R. By Remark I.1.1, **e** can be made into a basis element for **e**·R. Since **e**·R is a direct summand for R, this implies that **e** is also a basis element for R. Thus, since R splits into two nontrivial rings of rank one, Proposition V.1.1 gives either form 1 or 2, depending on whether **e** has additive order d_1 or d_2.

As to the last statement in the proposition, it

follows by direct computation.■

Because of the forms dictated by the existence of the nontrivial central idempotent, the number of isomorphism classes, given d_1 and d_2, is $d_1 + d_2 + 2$ when $d_1 > d_2$, and $d_1 + 1$ when the d_i are equal.

Wiesenbauer ([W3]) published a formula for counting the number of mutually nonisomorphic rank two rings with identity. Because of the simple structure of such rings, he was able to employ streamlined notation and analysis to arrive at this very excellent result.

5. Rings of rank two with noncentral idempotents.

The next two results are new, and with Proposition V.4.1 completely describe nonnilpotent rings without 1. However, the theorems do <u>not</u> positively identify mutually nonisomorphic forms.

Theorem V.1: Let R be of type $_p(d_1,d_2)$, e a noncentral idempotent of R, e·R a rank one ring. Then any cube [M] representing R is equivalent to $\begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix}$.

Proof: If R possesses a noncentral idempotent such that

e·R was a rank one ring, then any cube [M] representing R is equivalent to the form

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} a \\ b \end{bmatrix} & \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix}.$$

Associativity requires that $b^2 \equiv b \bmod p^{\alpha_2}$, $c \equiv 0 \bmod p^{\alpha_1}$, and $ab \equiv 0 \bmod p^{\alpha_2}$. This provides two possible cases:

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} a \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ d \end{bmatrix} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} a \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ d \end{bmatrix} \end{bmatrix}.$$

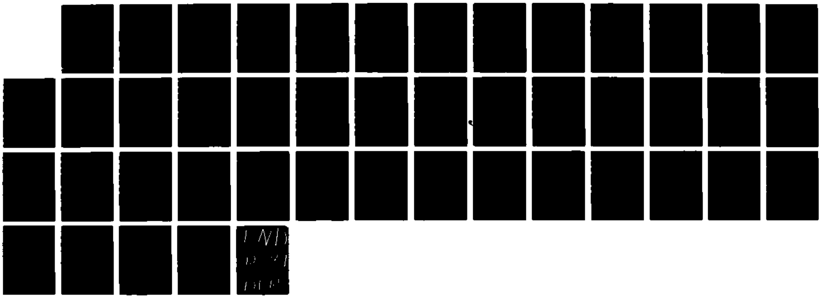
In the first case associativity forces $a \equiv 0 \bmod p^{\alpha_1}$, an impossibility since e is noncentral. This leaves us with the second case, which associativity forces both $a \equiv 0 \bmod p^{\alpha_1}$ and $d \equiv 0 \bmod p^{\alpha_2}$. This completes the proof of the theorem.■

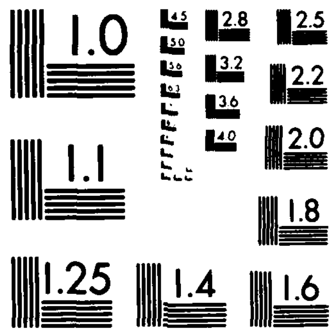
Theorem V.2: Let R be a ring of type $_p(\alpha_1,\alpha_2)$, e a noncentral idempotent of R, e·R a rank two ring. Then any cube [M] representing R is equivalent to one of the forms

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} p^{\epsilon_2} \\ 0 \end{bmatrix} & \begin{bmatrix} \beta p^{\epsilon_3} \\ \gamma p^{\delta_3} \end{bmatrix} \end{bmatrix} \quad \begin{cases} \alpha_1 \ge \epsilon_2, \epsilon_3 \ge \max\{\alpha_2, \alpha_1 - \alpha_2\} \\ \alpha_2 \ge \delta_3 \\ \beta, \gamma \in \mathbb{Z}/p \end{cases}$$

AD-A185 869 THE CLASSIFICATION PROBLEM OF FINITE RINGS BY COMPUTABLE MEANS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH W A KIELE 1987 2/2

UNCLASSIFIED AFIT/CI/NR-87-118T F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

or

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} p^{i_2} \\ 1 \end{bmatrix} & \begin{bmatrix} \beta p^{i_3} \\ \gamma p^{j_3} \end{bmatrix} \end{bmatrix} \quad \begin{cases} d_1 > i_2 \geq \max\{d_2, d_1 - d_2\} \\ d_1 \geq i_3 \geq d_1 - d_2 \\ d_2 \geq j_3 \\ \beta, \gamma \in \mathbb{Z}/p \end{cases} .$$

Proof: The idea follows exactly along the lines of the previous theorem. If R is a ring with noncentral idempotent e such that $e \cdot R = R$, then any cube representing R is equivalent to one of the form $\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} a \\ c \end{bmatrix} & \begin{bmatrix} b \\ d \end{bmatrix} \end{bmatrix}$, by Proposition

III.3.5. Associativity requires that

1) $b^2 \equiv b \mod p^{d_2}$,

2) $ab \equiv 0 \mod p^{d_2}$,

3) $bc \equiv c \mod p^{d_2}$,

4) $a + bd \equiv d \mod p^{d_2}$.

Thus by 1), b = 0 or 1, which gives two cases

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} a \\ 0 \end{bmatrix} & \begin{bmatrix} b \\ d \end{bmatrix} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} a \\ 1 \end{bmatrix} & \begin{bmatrix} b \\ d \end{bmatrix} \end{bmatrix}.$$

Case 1. b = 0 implies p^{d_2} divides c and a by 3) and 4).

Also, $a \equiv d \bmod p^{d_2}$, which means that for $d_1 = d_2$, we have a is a unit if and only if d is, and for $d_1 > d_2$, d is not a unit. Thus [M] is of the form

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} \alpha p^{i_2} \\ 0 \end{bmatrix} & \begin{bmatrix} \beta p^{i_3} \\ \delta p^{j_3} \end{bmatrix} \end{bmatrix} \begin{cases} d_1 \geq i_2, i_3 \geq \max\{d_2, d_1 - d_2\} \\ d_2 \geq j_3 \\ \alpha, \beta, \delta \in \mathbb{Z}/p \end{cases}.$$

Letting

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \alpha^{-1} \end{bmatrix},$$

we get

$$\mathcal{T}_A([M]) = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} p^{i_2} \\ 0 \end{bmatrix} & \begin{bmatrix} \beta p^{i_3} \\ \gamma p^{j_3} \end{bmatrix} \end{bmatrix} \begin{cases} d_1 \geq i_2, i_3 \geq \max\{d_2, d_1 - d_2\} \\ d_2 \geq j_3 \\ \beta, \gamma \in \mathbb{Z}/p \;, \; \gamma = \delta\alpha^{-1}. \end{cases} \square$$

Case 2. b = 1 implies that p^{d_2} divides a by 2). a cannot be 0 since e is noncentral. No other conclusions canbe drawn from associativity, and so [M] must be of the form

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} \alpha p^{i_2} \\ 1 \end{bmatrix} & \begin{bmatrix} \beta p^{i_3} \\ \delta p^{j_3} \end{bmatrix} \end{bmatrix} \begin{cases} d_1 > i_2 \geq \max\{d_2, d_1 - d_2\} \\ d_1 \geq i_3 \geq d_1 - d_2 \\ d_2 \geq j_3 \\ a, \beta, \gamma \in \mathbb{Z}/p \end{cases}.$$

Again letting

$A = \begin{bmatrix} 1 & 0 \\ 0 & \alpha^{-1} \end{bmatrix}$, we get

$$\mathcal{T}_A([M]) = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} p^{i_2} \\ 1 \end{bmatrix} \begin{bmatrix} \beta p^{i_3} \\ \gamma p^{j_3} \end{bmatrix} \end{bmatrix} \left\{ \begin{array}{l} d_1 > i_2 \geq \max\{d_2, d_1 - d_2\} \\ d_1 \geq i_3 \geq d_1 - d_2 \\ d_2 \geq j_3 \\ \beta, \gamma \in \mathbb{Z}/p \quad \gamma = \delta\alpha^{-1} \end{array} \right. .$$

This completes the proof of theorem.■

Theorems V.1 and V.2 place an upper bound on the possible number of mutually nonisomorphic rings. A closed form equation stating the exact number of isomorphism classes, given d_1 and d_2 would be a significant result on its own. The author hopes to be able to derive such a formula in future research.

CONCLUDING REMARKS

The results obtained in this paper give rise to three distinct avenues for further research which the author intends to pursue:

1. Nilpotent rings of rank 2 and nonnilpotent rings of rank 3. It is shown in [KP] that "most" rings are nilpotent; that is, as p or n gets larger, a higher percentage of the rings of cardinality p^n are nilpotent. In such rings, there are special elements such that $x^2 = p^i x$ -- p-potent elements, if you will. The author intends to explore if such elements can be used to classify nilpotent rings of rank 2. As for nonnilpotent rings of rank 3, many rings can already be identified from this paper's results; namely, those rings with a nontrivial central idempotent. The author hopes to classify all, or some special subset of rank 3 rings.

2. Improvements in the algorithms. The "linearization" of the quadratic identities is effective with rank 2 and some rank 3 three rings. However, it primarily serves to reduce the number of possible candidates among the transition matrices in testing for isomorphism between two cubes, Further, as was shown in Remark III.1.2, when considering characteristic 2^d rings, linearization is not effective in evaluating certain equations. Alternative means for testing these rings for isomorphism will be explored.

3. The application in algebraic cryptography alluded to in

Chapter II can be extensively developed, because the mapping $\mathcal{T}_A([M])$ exhibits many of the characteristics of a good encipherment system.

Given an enciphered cube of size k, the number of computations required, on average, to decipher a cube with k^3 elements is a function $k^n + f(k)$, $\deg f < n$, and would be considered a "hard" problem, but theoretically not impossible. For this reason, its discussion in depth is not considered appropriate here. However, a successful encipherment system need not have mathematically perfect security.

For example, one commercially successful public-key encryption system (RSA) depends on the high probability that there is no easy method of factoring a large (greater than 10^{100}) composite number, the product of two large (greater than 10^{50}) prime numbers. Currently known methods would require some 74 years. (See [MM] and [BP]).

By presenting $\mathcal{T}_A([M])$ as an encipherment system for publication in a periodical such as <u>Cryptologia</u>, it is hoped that others will attempt its solution, which will simultaneously provide us with effective means of classifying rings of larger rank. We refer the readers to the earlier discussion in Remark 2.3.

REFERENCES

[Ba]: Baumgartner, K. "Bemerkungen Zum Isomorphieproblem der Ringe", *Monatsch. Math.* 70 (1966),pp.299-308.

[Bo]: Turbo PASCAL™ is a trademark of Borland Int'l.

[BP]: Beker, Henry and Fred Piper, Cipher Systems, John Wiley & Sons, 1982.

[Ca]: Carmichael,A.D. Introduction to the Theory of Groups of Finite Order, Ginn and Co., 1937.

[Fu]: Fuchs, L. Infinite Abelian Groups, vol II, Academic Press, 1973.

[Ja]: Jacobsen, N. The Structure of Rings, American Math. Soc. Colloquium Publications, vol 37, 1964.

[KP]: Kruse, R.L. and David T. Price, Nilpotent Rings, Gordon and Breach Science Publishers, 1969.

[MM]: Meyer, Carl H. and Stephen M. Mattyas, Cryptography: A New Dimension in Computer Data Security, John Wiley & Sons, 1982.

[Ra]: Raghavendran, R. "Finite Associative Rings", *Compositio Mathematica*, Vol 21, Fasc. 2 (1969),pp. 195-229.

[Ro]: Roby, Norbert, "Sur le Cardinal du Group GL(n,A) où A est un Anneau Fini", *Anais Acad. Brazil C.*, 49 (1977). pp 15-18.

[To]: Toskey, B.R., "Rings on a Direct Sum of Cyclic Groups", *Publ Math Debrecen*, 10 (1963), pp. 93-95.

[W1]: Wiesenbauer, J. "Über die endlichen Ringe mit gegeben additiver Gruppe", *Monatsch. Math.* 78 (1974), pp. 164-173.

[W2]: ________, "Über die endlichen p-Ringe vom Rang Zwei", *Math. Balk.* 46 (1974), pp.723-725.

[W3]: ________ and Walter Flor, "Zum Klassificationsproblem endlicher Ringe",Osterreich Akad. Wiss.der Math-Natur Kl, Abt II, 183 no.8-10, pp.309-320.

```
Program Slicer;
{This program implements the basic group action described in
Chapter II of the dissertation.  All calculations, including matrix
inversion, is performed by integer arithmetic.  Each key Procedure
has a comment describing its purpose.}
{==========================Declarations============================}
Label
  FOUR;
Const
  Sz = 2;
  Base = 3;
Type
  Basis = array[1..Sz] of Integer;
  Matrix = array[1..Sz,1..Sz] of Integer;
  Cube = array[1..Sz,1..Sz,1..Sz] of Integer;
  Vec = array[0..1000] of Integer;
Var
  M,N: Cube;
  A,A1,Ainv: Matrix;
  I,J,L,Sing: Integer;
  Cbar:Vec;
  Bas:Basis;
{===========================Function Power=========================}
Function Power(P,v:Integer):Integer;
Begin
  Power:= Round(Exp(v*Ln(P)));
End;                                    {Function Power}
{==========================Procedure Inp===========================}
Procedure Inp(var Bas:Basis;var M:Cube;var A:Matrix);
Var
  I,J,K,Num: Integer;
Begin
 ClrScr;
 GotoXY(8,4);
 Writeln('R has rank ',Sz,' and P is ',Base,'.');
 GotoXY(8,8);
 Writeln('Enter the type of R in nonincreasing order.  Errors will');
 Writeln('not be detected, so be careful.');
 Writeln('IMPORTANT!! P**D1 Cannot Exceed 181, or else integer');
 Writeln(' overflow will occur, due to the language limitation.');
 GotoXY(1,12);
 For I:= 1 to Sz do begin
  Write('D',I,'=');
  Read(Kbd,Num);
  Bas[I]:= Num;
  Write(Bas[I]);
  GotoXY(1,WhereY+1);
 end;
 Writeln;
 Writeln('Input of Ring Type Complete.');
 Delay(3000);
 ClrScr;
```

```
 GotoXY(8,8);
 Writeln('Enter [M] by rows.  The entries will be reduced to the');
 Writeln('correct modulo if necessary.');
 GotoXY(1,10);
 For I:= 1 to Sz do begin
  For K:= 1 to Sz do begin
   For J:= 1 to Sz do begin
    Read(Kbd,Num);
    M[I,J,K]:= Num mod Power(Base,Bas[K]);
    GotoXY(5*J, WhereY);
    Write(M[I,J,K]);
   end;
   Writeln;
  end;
 end;
 GotoXY(1,20);
 Writeln('Input of Cube Complete.');
 Delay(3000);
 ClrScr;
 GotoXY(8,8);
 Writeln('Enter A by rows.  No error checking will be performed, so');
 Writeln('be sure that A is a transition matrix.');
 GotoXY(1,10);
 For I:= 1 to Sz do begin
  For J:= 1 to Sz do begin
   Read(Kbd,Num);
   A[I,J]:= Num;
   A1[I,J]:= Num;
   GotoXY(5*J,WhereY);
   Write(A[I,J]);
  end;
  Writeln;
 end;
 GotoXY(1,20);
 Writeln('Input of Transition Matrix Complete.');
 Delay(3000);
 ClrScr;
End;                                  {Procedure Inp}
{===========================Procedure Invint========================}
{Invint establishes a Look-Up table for rapid identification of zero
divisors and units, listing each unit's inverse.}
Procedure Invint(var Cbar: Vec;Bas:Basis);
Var
  I,J,Nmod: Integer;
Label
  ONE;
Begin
  Cbar[1]:=1;
  Nmod:= Power(Base,Bas[1]);
  For I:=2 to (Nmod-1) do begin
    If Cbar[I] = 0 then begin
      For J:=2 to (Nmod-1) do begin
```

```
        If (I*J) mod Nmod = 1 then begin
          Cbar[I]:= J;
          Cbar[J]:= I;
          Goto ONE;
        end;
      end;
    end;
  ONE:end;
                                   {Note that if inverse is not found, it
                                    is a zero divisor, and has already been
                                    initialized to zero.}
End;                                         {Procedure Invint}
{======================Procedure Invert===============================}
{Invert finds the inverse mod P**D1 of a matrix. If it is singular,
 the user will be advised and the program will be halted.}
Procedure Invert(var A1,Ainv:Matrix;Cbar:Vec; var Sing: Integer;
                 Bas:Basis);
Label
  TWO,THREE;
Var
  I,J,J1,J2,K,L1,L2,Piv,Temp,Mult,Nmod: Integer;
Begin
  Nmod:= Power(Base,Bas[1]);
  TWO:For I:= 1 to (Sz-1) do begin
    Piv:=I;
    THREE: If Cbar[A1[I,Piv]] = 0 then begin    {Pivot Search}
      Piv:=Piv + 1;
      If Piv > Sz then begin
        Writeln('Sorry. This matrix is singular.');
        Sing:=1;
        Exit;
      end;
      Goto THREE;
    end;                                     {Pivot Search}
    If I < Piv then begin                    {Row Swap}
      For J1:= 1 to Sz do begin
        Temp:= A1[I,J1];
        A1[I,J1]:= A1[Piv,J1];
        A1[Piv,J1]:= Temp;
        Temp:= Ainv[I,J1];
        Ainv[I,J1]:=Ainv[Piv,J1];
        Ainv[Piv,J1]:=Temp;
      end;
    end;                                     {Row Swap}
    For L1:= (I+1) to Sz do begin   {Upper Triangularization}
      Mult:=(A1[L1,I]*Cbar[A1[I,I]]) mod nmod;
      For L2:= I to Sz do begin
        A1[L1,L2]:= (A1[L1,L2] - Mult*A1[I,L2]) mod Nmod;
        If A1[L1,L2] < 0 then A1[L1,L2]:= A1[L1,L2] + Nmod;
        Ainv[L1,L2]:=(Ainv[L1,L2]-Mult*Ainv[I,L2]) mod Nmod;
        If Ainv[L1,L2] < 0 then Ainv[L1,L2]:= Ainv[L1,L2] + Nmod;
      end;
```

```
    end;
  end;                                    {Upper Triangularization}
  If Cbar[Al[Sz,Sz]] = 0  then begin
    Writeln('Sorry. Last element zero.');
    Sing:=1;
    Exit;
  end;
  For I:=Sz downto 2 do begin        {Diagonalization}
    For L1:= (I-1) downto 1 do begin
      Mult:= (Al[L1,I]*Cbar[Al[I,I]]) mod nmod;
      For L2:= I downto 1 do begin
        Al[L1,L2]:=(Al[L1,L2]-Mult*Al[I,l2]) mod nmod;
        If Al[L1,L2] < 0 then Al[L1,L2]:= Al[L1,L2] + Nmod;
        Ainv[L1,L2]:=(Ainv[L1,L2]-Mult*Ainv[I,L2]) mod nmod;
        If Ainv[L1,L2] < 0 then Ainv[L1,L2]:= Ainv[L1,L2] + Nmod;
      end;
    end;
  end;                                    {Diagonalization}
  For I:=1 to Sz do begin {Scaling of Diagonal Matrix to Identity}
    For J:=1 to Sz do begin
      Ainv[I,J]:=(Cbar[Al[I,I]]*Ainv[I,J]) mod Nmod;
    end;
  end;                                    {Scaling}
End;                                      {Procedure Invert}
{==========================Procedure Slicer=========================}
{This is the actual computation of the group action.}
Procedure Slicer(A,Ainv: Matrix;var M,N:Cube; Bas:Basis);
Var
  I,J,L,R,S,T,Nmod:Integer;
Begin
 For I:= 1 to Sz do begin
  For J:= 1 to Sz do begin
   For L:= 1 to Sz do begin
   Nmod:= Power(Base,Bas[L]);
    for R:= 1 to Sz do begin
     For S:= 1 to Sz do begin
      For T:= 1 to Sz do begin
       N[I,J,L]:=(N[I,J,L] + ((A[R,I]*A[S,J]) mod Nmod)*
       ((Ainv[L,T]*M[R,S,T]) mod Nmod)) mod Nmod;
      end;
     end;
    end;
    N[I,J,L]:=(N[I,J,L] + Nmod)mod nmod;
   end;
  end;
 end;
End;                                      {Procedure Slicer}
{============================Main Program============================}
Begin
{==========================Initialization==========================}
  Sing:=0;
  For I:=0 to 1000 do
```

```
    Cbar[I]:=0;
  For I:=1 to Sz do begin
    For J:=1 to Sz do begin
      A[I,J]:=0;
      A1[I,J]:=0;
      If I = J then
        Ainv[I,J]:=1
      else
        Ainv[I,J]:=0;
      For L:= 1 to Sz do begin
        M[I,J,L]:=0;
        N[I,J,L]:= 0;
      end;
    end;
  end;
{============================Execution==================================}
  Inp(Bas,M,A);
  Invint(Cbar,Bas);
  Invert(A1,Ainv,Cbar,Sing,Bas);      {Inverse of Slicer computed here}
  If Sing = 1 then goto FOUR;
  For I:=1 to Sz do begin
    For J:= 1 to Sz do
      Write(Lst,A[I,j]:5);
    Write(Lst,'     ');
    For J:= 1 to Sz do
      Write(Lst,Ainv[I,J]:5);
    Writeln(Lst);
  end;
  Slicer(A,Ainv,M,N,Bas);
  ClrScr;
  For I:= 1 to Sz do begin
    For L:= 1 to Sz do begin
      For J:= 1 to Sz do
        Write(M[I,J,L]:5);
      Write('     ');
      for J:= 1 to Sz do
        Write(N[I,J,L]:5);
      Writeln;
    end;
  end;
FOUR:End.                                  {Main Program}
```

```
Program BASPROPS;
{Three main procedures make up this program. The first, Multiplication,
 tests if the cube represents a multiplication.  The other two, titled
 Associativity and Commutativity, respectively, does what one expects.}
{==========================Declarations==============================}
Const
  Sz = 3;
  Base = 2;
Type
  Basis = array[1..Sz] of integer;
  Cube = array[1..Sz,1..Sz,1..Sz] of integer;
Var
  M: Cube;
  Bas: Basis;
  I,J,K: Integer;
{============================Function Power============================}
Function Power(B,V:Integer):Integer;
Begin
  Power:= Round(Exp(V*Ln(B)));
End;                                   {Function Power}
{===========================Procedure Inp============================}
Procedure Inp(var Bas:Basis;var M:Cube);
var
  I,J,K, Num:Integer;
Begin
  ClrScr;
  Writeln('Ring is of rank ',Sz,' and P is ',Base,'.');
  Writeln('To change, edit program constants Sz and Base.');
  GotoXY(8,3);
  Writeln('Enter the type of R in nonincreasing order.  Errors will');
  Writeln('be detected, so be careful.');
  GotoXY(1,5);
  For I:= 1 to Sz do begin
    Write('D',I,'=');
    Read(Kbd,Num);
    Bas[I]:= Num;
    Write(Bas[I]);
    GotoXY(1,WhereY+1);
  end;
  Writeln('Input of Ring Type Complete.');
  Delay(3000);
  ClrScr;
  GotoXY(8,8);
  Writeln('Enter [M] by rows.  The entries will be reduced if to the');
  Writeln('correct modulo if necessary.');
  GotoXY(1,10);
  For I:= 1 to Sz do begin
    For K:= 1 to Sz do begin
      For J:= 1 to Sz do begin
        Read(Kbd,Num);
        M[I,J,K]:= Num mod(Power(Base,Bas[K]));
        GotoXY(5*J,WhereY);
```

```
          Write(M[I,J,K]);
        end;
        Writeln;
      end;
    end;
    GotoXY(1,20);
    Writeln('Input of Cube Complete.');
    Delay(3000);
  End;                                     {Procedure Inp}
  {============================Procedure Multiplication=================}
  Procedure Multiplication(Bas:Basis;M:Cube);
  Var
   I,J,K:Integer;
  Begin
    If Bas[1] = Bas[Sz] then begin
      Writeln('R is a free module; hence [M] is a multiplication.');
      Exit;
    end;
    For I:= 1 to Sz do begin
      For J:= 1 to Sz do begin
        For K:= 1 to Sz do begin
          If (Bas[K] > Bas[J]) and (Bas[I] >= Bas[J]) then begin
            If M[I,J,K] mod(Power(Base,(Bas[K]-Bas[J]))) <> 0 then begin
              Writeln('[M] is not a multiplication.');
              Exit;
            end;
          end;
          If (Bas[K] > Bas[I]) and (Bas[J] >= Bas[I]) then begin
            If M[I,J,K] mod(Power(Base,(Bas[K]-Bas[I]))) <> 0 then begin
              Writeln('[M] is not a multiplication.');
              Exit;
            end;
          end;
        end;
      end;
    end;
    Writeln('[M] is a multiplication.');
  End;                                     {Procedure Multiplication}
  {============================Procedure Associativity==================}
  Procedure Associativity(Bas:Basis;M:Cube);
  Var
    C,I,J,T,U,SUM,NMOD: Integer;
  Begin
   For C:= 1 to Sz do begin            {Bas[C]}
   Nmod:= Power(Base,Bas[C]);
    For I:= 1 to Sz do begin
     For J:= 1 to Sz do begin
      For T:= 1 to Sz do begin
       Sum:= 0;
       For U:= 1 to Sz do begin
        Sum:=(Sum + M[I,J,U]*M[U,T,C] - M[J,T,U]*M[I,U,C]) mod
        Power(Base,Bas[C]);
```

```
      end;
      If (Sum + Nmod) mod Nmod <> 0 then begin
       Writeln('[M] is not associative.');
       Exit;
      end;
     end;
    end;
   end;
  end;
  Writeln('[M] is associative.');
 End;                                    {Procedure Associativity}
 {===========================Procedure Commutativity=================}
 Procedure Commutativity(Bas:Basis;M:Cube);
 Var
  C,I,J: Integer;
 Begin
  For C:= 1 to Sz do begin
   For I:= 1 to Sz do begin
    For J:= 1 to Sz do begin
     If (M[I,J,C] - M[J,I,C]) <> 0 mod(Power(Base,Bas[C])) then begin
      Writeln('[M] is not commutative.');
      Exit;
     end;
    end;
   end;
  end;
  Writeln('[M] is commutative.');
 End;                                    {Procedure Commutativity}
 {============================Main Program============================}
 Begin
 {==========================Initialization==========================}
  For I:= 1 to Sz do begin
   For J:= 1 to Sz do begin
    For K:= 1 to Sz do begin
     M[I,J,K]:= 0;
    end;
   end;
   Bas[I]:=0;
  end;
 {===========================Execution===============================}
  Inp(Bas,M);
  Multiplication(Bas,M);
  Associativity(Bas,M);
  Commutativity(Bas,M);
 End.                                    {Main Program}
```

```
Program Identity;
{This Program takes a cube and checks it for the existence of an
identity, implementing Equation III.1.9 of the dissertation.  To
evaluate rings of rank > 5, change Siz and Max as indicated by
comments in Declarations.}
{==========================Declarations=============================}
Const
  Sz = 2;
  Siz = 6;                              {Sz + 1}
  Base = 5;
  Nmod = 125;
  Max = 50;                             {2*Sz**2}
Type
  Basis = array[1..Sz] of Integer;
  Blk = array[0..1000] of Integer;
  Cube = array[1..sz,1..sz,1..sz] of Integer;
  Vec = array[0..1000,1..2] of Integer;
  Matrix = array[1..Max,1..5] of Integer;
Var
  Bas: Basis;
  I,J,K,K1: Integer;
  M: Cube;
  Cbar: Blk;
  Padic: Vec;
  T1,T2: Matrix;
{==========================Function Power===========================}
Function Power( B,v:Integer):Integer;
Begin
  Power:=Round(Exp(V*Ln(B)));
end;                                    {Function Power}
{==========================Procedure Inp============================}
Procedure Inp(var Bas:Basis;var M:Cube);
var
  I,J,K, Num:Integer;
Begin
  ClrScr;
  Writeln('Ring is of rank ',Sz,' and P is ',Base,'.');
  Writeln('To change, edit program constants Sz and Base.');
  GotoXY(8,3);
  Writeln('Enter the type of R in nonincreasing order.  Errors will');
  Writeln('be detected, so be careful.');
  GotoXY(1,5);
  For I:= 1 to Sz do begin
    Write('D',I,'=');
    Read(Kbd,Num);
    Bas[I]:= Num;
    Write(Bas[I]);
    GotoXY(1,WhereY+1);
  end;
  Writeln('Input of Ring Type Complete.');
  Delay(3000);
  ClrScr;
```

```
  GotoXY(8,8);
  Writeln('Enter [M] by rows.  The entries will be reduced if to the');
  Writeln('correct modulo if necessary.');
  GotoXY(1,10);
  For I:= 1 to Sz do begin
    For K:= 1 to Sz do begin
      For J:= 1 to Sz do begin
        Read(Kbd,Num);
        M[I,J,K]:= Num mod(Power(Base,Bas[K]));
        GotoXY(5*J,WhereY);
        Write(M[I,J,K]);
      end;
      Writeln;
    end;
  end;
  GotoXY(1,20);
  Writeln('Input of Cube Complete.');
  Delay(3000);
End;                                   {Procedure Inp}
{==========================Procedure Invint========================}
Procedure Invint (var cbar: Blk);
Var
  I,J: Integer;
Label
  ONE;
Begin
  Cbar[1]:= 1;
  For I:= 2 to (Nmod - 1) do begin
    If Cbar[I] = 0 then begin
      For J:= 2 to (Nmod - 1) do begin
        If (I*J) mod Nmod = 1 then begin
          Cbar[I]:= J;
          Cbar[J]:= I;
          Goto ONE;
        end;
      end;
    end;
  ONE:end;
end;                                   {Procedure Invint}
{==========================Procedure Invval========================}
{If X = (P**K)*U, then K = Padic[X,1] and U = Padic[X,2].  If X = 0,
then k = Expo.}
Procedure Invval(var Padic:Vec);
Var
  K,I,Expo: Integer;
Begin
  Expo:= Round(Ln(Nmod)/Ln(Base));
  Padic[0,1]:=Expo;
  For I:= 1 to (Nmod-1) do begin
    K:=1;
    While (I mod Power(Base,K) = 0) and (K <= Expo) do
      K:= K+1;
```

```
    Padic[I,1]:= K-1;
    Padic[I,2]:= I div Power(Base,(K-1));
  end;
end;                                     {Procedure Invval}
{===========================Procedure Matrixmaker====================}
Procedure Matrixmaker(var T1,T2:Matrix;M:Cube;Bas:Basis);
Var
 I,J,K,Sum,Row1,Row2,Col1,Col2: Integer;
Begin
 For I:= 1 to Sz do begin
  For J:= 1 to Sz do begin
   For K:= 1 to Sz do begin
    Row1:= I + Sz*(K-1);
    Col1:= J;
    Row2:= Power(Sz,2) + J + Sz*(K-1);
    Col2:= I;
    T1[Row1,Col1]:= M[I,J,K]*Power(Base,(Bas[1]-Bas[K]));
    T1[Row2,Col2]:= M[I,J,K]*Power(Base,(Bas[1]-Bas[K]));
    T2[Row1,Col1]:= M[I,J,K]*Power(Base,(Bas[1]-Bas[K]));
    T2[Row2,Col2]:= M[I,J,K]*Power(Base,(Bas[1]-Bas[K]));
   end;
  end;
 end;
End;                                     {Procedure Matrixmaker}
{============================Procedure Reduce========================}
Procedure Reduce (var T2:Matrix; cbar: Blk; Padic:vec);
Label
  ONE,TWO,THR;
Var
I,I3,I5,J,J3,J5,Piv,J1,L1,L2,Mult,Bot,Temp,Minpower,Minrow,
  Diff,Expo:Integer;
Begin
  Expo:= Round(Ln(Nmod)/Ln(Base));
  Bot:=2*Power(Sz,2);
  J:=0;
  For I:= 1 to (Sz+1) do begin         {Loop}
    TWO:If J <= Sz then
      J:=J+1
    else
      Goto THR;
    Piv:= I;
    Minpower:= Expo;
    Minrow:= I;
    While(Padic[T2[Piv,J],1] >= 1) and (piv <= Bot) and (J<= (Sz+1))do
    begin                                {Pivot Search}
      If (Piv = I) then
        Minpower:= Padic[T2[Piv,J],1];
      If Piv > I then begin
        If Padic[T2[Piv,J],1] < Padic[T2[(Piv-1),J],1] then begin
          Minpower:= Padic[T2[Piv,J],1];
          Minrow:= Piv;
        end;
```

```
      end;
      Piv:= Piv + 1;
    end;                                    {Pivot Search}
    If (Piv > Bot) and (Minpower=Expo) then
      Goto TWO;
    If (Piv > Bot) and (Minpower < Expo) then
      Piv:= Minrow;
    If I < Piv then begin                   {Pivot}
      For J1:= 1 to (Sz+1) do begin
        Temp:= T2[I,J1];
        T2[I,J1]:=T2[Piv,J1];
        T2[Piv,J1]:= Temp;
      end;
    end;                                    {Pivot}
    THR:For J3:= 1 to (Sz+1) do begin
      If T2[I,J3]<>0 then begin
        Mult:= Cbar[Padic[T2[I,J3],2]];
        For J1:=J3 to (Sz+1) do
          T2[I,J1]:= (Mult*T2[I,J1])mod Nmod;
        Goto ONE;
      end;
    end;
    ONE:For L1:= (I + 1) to Bot do begin    {Reduce Bot L1}
      Diff:=Power(Base,(Abs(Padic[T2[L1,J],1]-Padic[T2[I,J],1])));
      Mult:=((Diff*Cbar[Padic[T2[I,J],2]]mod Nmod)
           *Padic[T2[L1,J],2]) mod Nmod;
      For L2:= J to (Sz+1) do
        T2[L1,L2]:=(((T2[L1,L2]-Mult*T2[I,L2])mod Nmod)+Nmod)mod Nmod;
    end;                                    {Reduce Bot L1}
    For I5:=1 to 2*Power(Sz,2) do begin
        For J5:= 1 to (Sz+1)do
          Write(Lst,T2[I5,J5]:4);
        Writeln(Lst);
      end;
  end;                                      {Loop}
end;                                        {Procedure Reduce}
{===========================Main Program============================}
Begin
{===========================Initialization==========================}
  For I:= 0 to Nmod do begin
    Cbar[I]:= 0;
    Padic[I,1]:=0;
    Padic[I,2]:=0;
  end;
  For I:= 1 to Sz do begin
    Bas[I]:= 0;
    For J:= 1 to Sz do begin
      For K:= 1 to Sz do begin
        M[I,J,K]:= 0;
      end;
    end;
  end;
```

```
{==========================Execution================================}
  Inp(Bas,M);
  For I:= 1 to (2*Power(Sz,2)) do begin
    For J:= 1 to (Sz+1) do begin
      T1[I,J]:=0;
      T2[I,J]:=0;
    end;
    For K:= 0 to (Sz-1) do begin
      K1:= 1+K*(Sz+1);
      T1[K1,(Sz+1)]:= Power(Base,(Bas[1]-Bas[K+1]));
      T2[K1,(Sz+1)]:= Power(Base,(Bas[1]-Bas[K+1]));
      T1[(K1+Power(Sz,2)),(Sz+1)]:= Power(Base,(Bas[1]-Bas[K+1]));
      T2[(K1+Power(Sz,2)),(Sz+1)]:= Power(Base,(Bas[1]-Bas[K+1]));
    end;
  end;
  Invint(Cbar);
  Invval(Padic);
  Matrixmaker(T1,T2,M,Bas);
  Reduce(T2,Cbar,Padic);
  For I:=1 to 2*Power(Sz,2) do begin
    For J:= 1 to (Sz+1)do
      Write(Lst,T1[I,J]:4);
    Write(Lst,'              ');
    For J:=1 to (Sz+1) do
      Write(Lst,T2[I,J]:4);
    Writeln(Lst);
  end;
End.                                    {Program}
```

```
Program Combo;
{This Program takes two cubes, evaluates their traces, sets up the
appropriate linear systems involving the Trace Formulas, and reduces
the systems to echelon form. The output will be used as follows:
1) If inconsistent, the two cubes represent totally inequivalent rings.
2) If consistent, use the output in the next step in order to solve
   the system of quadratic equations which define the ring isomorphism,
   should one exist.}
{==========================Declarations============================}
Label
 REP,REP2,LOOPI,LOOPI2;
Const
  Sz = 2;
  Nmod = 3;
  Base = 3;
Type
  Blk = array[0..Nmod] of Integer;
  Sys = array[1..54,1..55] of integer;
  Cube = array[1..sz,1..sz,1..sz] of Integer;
  Vec = array[0..Nmod,1..2]of Integer;
  LftSd = Array[1..Sz,1..Sz,1..Sz,1..Sz,1..Sz] of Integer;
  Rtsd = Array[1..Sz,1..Sz,1..Sz,1..Sz] of Integer;
  Stak = Array[1..136] of Integer;
Var
  I,J,J3,K,U,V,rstep1,rstep2,cstep1,cstep2,Flag1, Flag2,Runnum:Integer;
  Sys1,Sys2,Raw,Work,Trace: Sys;
  M,N: Cube;
  Cbar: Blk;
  Padic: Vec;
  Lft1,Lft2: LftSd;
  Rt1,Rt2: Rtsd;
  Chol: Stak;
{==========================Function Power==========================}
Function Power( B,v:Integer):Integer;
Begin
  Power:=Round(Exp(V*Ln(B)));
end;                                 {Function Power}
{==========================Procedure Inp===========================}
Procedure Inp (var M,N:Cube);
  Var I,J,K,Num: Integer;
begin
  ClrScr;
  GotoXY(8,1);
  Writeln('Char R is ',Nmod,'; its rank is ',Sz,'; P is ',Base,'.');
  Writeln('They can be changed by editing the program constants.');
  GotoXY(8,4);
  Writeln('Input row of [M], then row of [N].');
  GotoXY(1,6);
  For I:= 1 to Sz do begin
    For K:= 1 to Sz do begin
      For J:= 1 to Sz do begin
        Read(Kbd,Num);
```

```
        M[I,J,K]:= Num;
        GotoXY((3*J),WhereY);
        Write(Num);
      end;
      For J:= 1 to Sz do begin
        Read(Kbd,Num);
        N[I,J,K]:= Num;
        GotoXY((3*J +3*Sz+5),WhereY);
        Write(Num);
      end;
      Writeln;
    end;
  end;
  GotoXY(1,23);
  Write(' Input of [M] and [N] is complete.');
end;                                    {Procedure Inp}
{===========================Procedure Tracer=======================}
Procedure Tracer(M,N:Cube; var Trace:Sys);
Var
  Sum1_,Sum_1,Sum2_,Sum_2,I,J: Integer;
begin
  For I:= 1 to Sz do begin
    Sum1_:= 0;
    Sum_1:= 0;
    Sum2_:= 0;
    Sum_2:= 0;
    For J:= 1 to Sz do begin
      Sum1_:=Sum1_ + M[I,J,J];
      Sum_1:=Sum_1 + M[J,I,J];
      Sum2_:=Sum2_ + N[I,J,J];
      Sum_2:=Sum_2 + N[J,I,J];
    end;
    Trace[I,5]:=Sum1_ mod Nmod;
    Trace[5,I]:=Sum_1 mod Nmod;
    Trace[I,6]:=Sum2_ mod Nmod;
    Trace[6,I]:=Sum_2 mod Nmod;
  end;
end;                                    {Procedure Trace}
{===========================Procedure SysBuilder====================}
Procedure SysBuilder(var Sys1,Sys2:sys; Trace:Sys);
Var
  I,J: Integer;
begin
  For I:= 1 to (2*Sz) do begin         {Loop}
    If I mod 2 = 1 then begin          {When I is Odd}
      For J:= (Sz*(I div 2)+1) to (Sz*((I div 2)+1)) do begin
        Sys1[I,J]:= Trace[(((J-1)mod sz)+1),5];
        Sys2[I,J]:= Trace[(((J-1)mod Sz)+1),6];
      end;
      Sys1[I,(Sz*Sz + 1)]:= Trace[((I + 1)div 2),6];
      Sys2[I,(Sz*Sz + 1)]:= Trace[((I + 1)div 2),5];
    end                                {When I is Odd}
```

```
    else begin                               {When I is Even}
      For J:= (Sz*((I-1) div 2)+1) to (Sz*(((I-1) div 2)+1)) do begin
        Sys1[I,J]:= Trace[5,(((J-1)mod Sz)+1)];
        Sys2[I,J]:= Trace[6,(((J-1)mod Sz)+1)];
      end;
      Sys1[I,(Sz*Sz + 1)]:= Trace[6,I div 2];
      Sys2[I,(Sz*Sz + 1)]:= Trace[5,I div 2];
    end;                                     {When I is even}
  end;                                       {Loop}
end;                                         {Procedure SysBuilder}
{===========================Procedure Invint========================}
Procedure Invint (var cbar: Blk);
Var
  I,J: Integer;
Label
  ONE;
Begin
  Cbar[1]:= 1;
  For I:= 2 to (Nmod - 1) do begin
    If Cbar[I] = 0 then begin
      For J:= 2 to (Nmod - 1) do begin
        If (I*J) mod Nmod = 1 then begin
          Cbar[I]:= J;
          Cbar[J]:= I;
          Goto ONE;
        end;
      end;
    end;
  ONE:end;
end;                                         {Procedure Invint}
{===========================Procedure Invval========================}
{If X = (P**K)*U, then Padic[X,1] = K, Padic[X,2] = U.  If X = 0, then
 Padic[X,2] = Expo.}
Procedure Invval(var Padic:Vec);
Var
  K,I,Expo: Integer;
Begin
  Expo:= Round(Ln(Nmod)/Ln(Base));
  Padic[0,1]:=Expo;
  For I:= 1 to (Nmod-1) do begin
    K:= 1;
    While (I mod Power(Base,K) = 0) and (K <= Expo) do
      K:= K+1;
    Padic[I,1]:= K-1;
    Padic[I,2]:= I div Power(Base,(K-1));
  end;
end;                                         {Procedure Invval}
{===========================Procedure LeftSide======================}
Procedure LeftSide(var Lft1,Lft2: LftSd; M: Cube);
Var
  I,L,C,U,V: Integer;
Begin
```

```
  For L:= 1 to Sz do begin
    For V:= 1 to Sz do begin
      For I:= 1 to Sz do begin
        For U:= 1 to Sz do begin
          For C:= 1 to Sz do begin
            Lft1[L,V,I,U,C]:= M[V,U,L];
            Lft2[L,V,I,U,C]:= M[U,V,L];
          end;
        end;
      end;
    end;
  end;
end;                                    {LeftSide}
{===========================Procedure RightSide======================}
Procedure RightSide(var Rt1,Rt2:Rtsd; N: Cube);
Var
  I,L,C,U: Integer;
Begin
  For I:= 1 to Sz do begin
    For L:= 1 to Sz do begin
      For C:= 1 to Sz do begin
        For U:= 1 to Sz do begin
          Rt1[I,L,C,U]:=N[I,C,U];
          Rt2[I,L,C,U]:=N[C,I,U];
        end;
      end;
    end;
  end;
end;                                    {Procedure RightSide}
{===========================Procedure Coltracker=====================}
{Arranges quadratic variables in lexicographic order.}
Procedure ColTracker (var Chol: Stak);
Var
  Knt,I,U,C,V: Integer;
Begin
  Knt:=0;
  For V:= 1 to Sz do begin
    For I:= 1 to Sz do begin
      For U:= 1 to Sz do begin
        For C:= 1 to Sz do begin
          If (10*V+I)<=(10*U+C) then begin
            Knt:= Knt + 1;
            Chol[Knt]:= C + 10*(U + 10*(I + 10*V));
          end;
        end;
      end;
    end;
  end;
End;                                    {Procedure Coltracker}
{===========================Procedure BigTableau====================}
Procedure BigTableau (var Raw,Work:Sys; Rt1,Rt2:Rtsd; Lft1,Lft2:LftSd;
                      Chol:Stak);
```

```
Var
  C,I,L,U,V,Check1,Check2,Ro,Nro,Kol,QuadCols,J,LastCol: Integer;
Label
  ONE,TWO;
Begin
  QuadCols:=(Power(Sz,4)+Power(Sz,2))div 2;
  LastCol:= QuadCols + Power(Sz,2) + 1;
  For L:= 1 to Sz do begin               {Entries from LeftSide}
   For V:= 1 to Sz do begin
    For I:= 1 to Sz do begin
     For U:= 1 to Sz do begin
      For C:= 1 to Sz do begin
       Ro:= C+Sz*((L-1)+Sz*(I-1));
       Check1:=C+10*(U+10*(I+10*V));
       Check2:=I+10*(V+10*(C+10*U));
       For J:= 1 to QuadCols do begin
        If(Chol[J]=Check1) or (Chol[J]=Check2) then begin
         Raw[Ro,J]:=(Raw[Ro,J]+Lft1[L,V,I,U,C])mod Nmod;
         Work[Ro,J]:= Raw[Ro,J];
         Nro:= Ro + Power(Sz,3);
         Raw[Nro,J]:=(Raw[Nro,J]+Lft2[L,V,I,U,C])mod Nmod;
         Work[Nro,J]:= Raw[Nro,J];
         Goto TWO;
        end;
       end;                              {j}
      TWO:end;                           {c}
     end;                                {u}
    end;                                 {i}
   end;                                  {v}
  end;                                   {Entries from LeftSide}
  For I:= 1 to Sz do begin               {Entries from RightSide}
   For L:= 1 to Sz do begin
    For C:= 1 to Sz do begin
     For U:= 1 to Sz do begin
      Ro:= C+Sz*((L-1)+Sz*(I-1));
      Kol:= QuadCols+U+(L-1)*Sz;
      Raw[Ro,Kol]:= (Nmod-Rt1[I,L,C,U])mod Nmod;
      Work[Ro,Kol]:= (Nmod-Rt1[I,L,C,U])mod Nmod;
      Nro:= Ro + Power(Sz,3);
      Raw[Nro,Kol]:= (Nmod-Rt2[I,L,C,U])mod Nmod;
      Work[Nro,Kol]:= (Nmod-Rt2[I,L,C,U])mod Nmod;
     end;
    end;
   end;
  end;                                   {Entries from RightSide}
End;                                     {Procedure BigTableau}
{===========================Procedure Reduce========================}
Procedure Reduce (var A:sys;cbar:Blk;Bot,LastCol:Integer;Padic:vec);
Label
  ONE,TWO,THR;
Var
  I,I3,J,J3,Piv,J1,L1,L2,Diff,Mult,Temp,Minpower,Minrow,Expo: Integer;
```

```
Begin
  Expo:= Round(Ln(Nmod)/Ln(Base));
  J:=0;
  ClrScr;
  For I:= 1 to (Bot -1)do begin          {Loop}
    TWO:If J < Lastcol then
      J:=J+1
    else
      Goto THR;
    Piv:= I;
    Minpower:= Expo;
    Minrow:= I;
    While(Padic[A[Piv,J],1] >= 1) and (piv <= Bot) and (J<= LastCol) do
    begin                                 {Pivot Search}
      If (Piv = I) then
        Minpower:= Padic[A[Piv,J],1];
      If Piv > I then begin
        If Padic[A[Piv,J],1] < Padic[A[(Piv-1),J],1] then begin
          Minpower:= Padic[A[Piv,J],1];
          Minrow:= Piv;
        end;
      end;
      Piv:= Piv + 1;
    end;                                  {Pivot Search}
    If (Piv > Bot) and (Minpower=Expo) then
      Goto TWO;
    If (Piv > Bot) and (Minpower < Expo) then
      Piv:= Minrow;
    If I < Piv then begin                 {Pivot}
      For J1:= 1 to LastCol do begin
        Temp:= A[I,J1];
        A[I,J1]:=A[Piv,J1];
        A[Piv,J1]:= Temp;
      end;
    end;                                  {Pivot}
    For L1:= (I + 1) to Bot do begin       {Reduce Bot L1}
      Diff:= Power(Base,(Abs(Padic[A[L1,J],1]-Padic[A[I,J],1])));
      Mult:=((Diff*Cbar[Padic[A[I,J],2]] mod Nmod)
            *Padic[A[L1,J],2])mod Nmod;
      For L2:= J to LastCol do
        A[L1,L2]:=((A[L1,L2]-((Mult*A[I,L2]) mod Nmod))+ Nmod)mod Nmod;
    end;                                  {Reduce Bot L1}
  end;                                    {Loop}
  THR:For I3:= 1 to Bot do begin
    For J3:= 1 to LastCol do begin
      If A[I3,J3] <> 0 then begin
        Mult:= Cbar[Padic[A[I3,J3],2]];
        For J1:=J3 to LastCol do
          A[I3,J1]:= (Mult*A[I3,J1])mod Nmod;
        Goto ONE;
      end;
    end;
```

```
  ONE:end;
end;                                    {Procedure Reduce}
{========================Procedure BTDep========================}
Procedure BTDep(var A:Sys; Chol:Stak);
Label
  LOOPC;
Var
 I2,J2,D1,D2,QuadCols,Bot,LastCol,Kol,Dest,Sig,Tau,
 Num,J,K,U,C,Ro,Check1,Check2: Integer;
Begin
 Bot:= 2*Power(Sz,3);
 QuadCols:= (Power(Sz,4)+Power(Sz,2))div 2;
 LastCol:= Quadcols + Power(Sz,2) + 1;
 ClrScr; GotoXY(1,8);
 Writeln('Modding Big Tableau. Assume you know
          A(Sig,Tau) = D1-D2*A(I2,J2).');
 Write ('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
 Write ('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
 Write ('I2 = ');   Read(Num);   I2:= Num;   Writeln;
 Write ('J2 = ');   Read(Num);   J2:= Num;   Writeln;
 Write ('D1 = ');   Read(Num);   D1:= Num;   Writeln;
 Write ('D2 = ');   Read(Num);   D2:= Num;   Writeln;
 Writeln('Working.');
 Kol:= QuadCols + Tau + Sz*(Sig-1);
 For U:= 1 to Sz do begin
  For C:= 1 to Sz do begin
   For J:= 1 to QuadCols do begin
    Check1:=C+10*(U+10*(Tau+10*Sig));
    Check2:=Tau+10*(Sig+10*(C+10*U));
    If(Chol[J]=Check1) or (Chol[J]=Check2) then begin
     Dest:= QuadCols + C + Sz*(U-1);
     For Ro:= 1 to Bot do
      A[Ro,Dest]:=(A[Ro,Dest]+D1*A[Ro,J])mod Nmod;
     For K:= 1 to QuadCols do begin
      Check1:=C+10*(U+10*(J2+10*I2));
      Check2:=J2+10*(I2+10*(C+10*U));
      If(Chol[K]=Check1) or (Chol[K]=Check2) then begin
       For Ro:= 1 to Bot do begin
        A[Ro,K]:=((A[Ro,K]-D2*A[Ro,J])mod Nmod + Nmod)mod Nmod;
        A[Ro,J]:=0;
       end;
       Goto LOOPC;
      end;          {k}
     end;           {Ro}
    end;            {Check#}
   end;             {j}
  LOOPC:end;
 end;
End;                                    {Procedure BTDep}
{========================Procedure BTCon========================}
Procedure BTCon(var A:Sys; Chol:Stak);
Label
```

```
 LOOPC;
Var
 QuadCols,Bot,LastCol,Kol,Dest,Sig,Tau,
 Num,Con,J,U,C,Ro,Check1,Check2: Integer;
Begin
 Bot:= 2*Power(Sz,3);
 QuadCols:= (Power(Sz,4)+Power(Sz,2))div 2;
 LastCol:= Quadcols + Power(Sz,2) + 1;
 ClrScr; GotoXY(1,8);
 Writeln('Modding Big Tableau.  Assume you know A(Sig,Tau) = Con.');
 Write ('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
 Write ('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
 Write ('Con = ');   Read(Num);   Con:= Num;   Writeln;
 Writeln('Working.');
 Kol:= QuadCols + Tau + Sz*(Sig-1);
 For U:= 1 to Sz do begin
  For C:= 1 to Sz do begin
   For J:= 1 to QuadCols do begin
    Check1:= C+10*(U+10*(Tau+10*Sig));
    Check2:= Tau+10*(Sig+10*(C+10*U));
    If(Chol[J]=Check1) or (Chol[J]=Check2) then begin
     Dest:= QuadCols + C + Sz*(U-1);
     For Ro:= 1 to Bot do begin
      A[Ro,Dest]:=(A[Ro,Dest]+Con*A[Ro,J])mod Nmod;
      A[Ro,J]:=0;
     end;
     Goto LOOPC;
    end;
   end;
  LOOPC:end;
 end;
 For Ro:= 1 to Bot do begin
  A[Ro,LastCol]:=((Nmod-Con*A[Ro,Kol])mod Nmod + Nmod)mod Nmod;
  A[Ro,Kol]:=0;
 end;
End;                                   {Procedure BTCon}
{===========================Procedure BTScreen=======================}
Procedure BTScreen(var Work:Sys; var Flag2: Integer; Chol:Stak);
Label
 REP;
Var
  Ch:Char;
Begin
  REP:ClrScr;
  GotoXY(1,8);
  Writeln('Choose one of the following options:');
  Writeln('Change Variable to [C]onstant.');
  Writeln('Re-express [D]ependent Variable in terms of one other.');
  Writeln('[R]educe Big Tableau again.');
  Writeln('[Q]uit Modifying Big Tableau.');
  Repeat
    Read(Kbd,Ch)
```

```
  Until Upcase(Ch) in ['C','D','R','Q'];
  Case Upcase(Ch) of
    'C': begin
           BTCon(Work,Chol);
           Goto REP;
         end;
    'R': Flag2:= 1;
    'D': begin
           BTDep(Work,Chol);
           Goto REP;
         end;
    'Q': Exit;                          {To Iterate}
  end;
End;                                    {Procedure BTScreen}
{===========================Procedure TF============================}
Procedure TF(var Sys1:Sys);
Var
  Bot,Col,Num,K,Sig,Tau,Con,I: Integer;
Begin
  Bot:=2*Sz;
  Col:= Power(Sz,2)+1;
  ClrScr;
  GotoXY(1,8);
  Writeln('Modifying Trace Formula. Assumes you know A(Sig,Tau)=Con.');
  Write('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
  Write('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
  Write('Con = ');   Read(Num);   Con:= Num;   Writeln;
  Writeln('Working.');
  K:=Sz*(Sig-1) + Tau;
  For I:= 1 to Bot do begin
    Sys1[I,Col]:= ((Sys1[I,Col]-Con*Sys1[I,K])mod Nmod + Nmod)mod Nmod;
    Sys1[I,K]:=0;
  end;
End;                                    {Procedure TF}
{===========================Procedure TFScreen======================}
Procedure TFScreen(var Sys1:Sys; var Flag1: Integer);
Label
  ONE;
Var
  Ch: Char;
Begin
  ONE:ClrScr;
  GotoXY(8,8);
  Writeln('Choose one of the following options:');
  Writeln('Change Variable to [C]onstant.');
  Writeln('[R]educe Trace Formula Again.');
  Writeln('[Q]uit.');
  GotoXY(8,10);
  Repeat
    Read(Kbd,Ch);
  Until Upcase(Ch) in ['C','R','Q'];
  Case Upcase(Ch) of
```

```
    'C': begin
           TF(Sys1);
           Goto ONE;
         end;
    'Q': Exit;                           {To Iterate}
    'R': Flag1:=1;
  end;
End;                                     {Procedure TFScreen}

{===========================Procedure Iterate=========================}
Procedure Iterate(var Sys1,Work:Sys;var Flag1,Flag2:Integer;Chol:Stak);
Var
  Ch: Char;
Begin
  ClrScr;
  GotoXY(10,8);
  Writeln('Modify [T]race Formulas, [B]ig Tableau, or
           [Q]uit Modifying.');
  GotoXY(10,10);
  Repeat
    Read(Kbd,Ch)
  Until Upcase(Ch) in ['T','B','Q'];
  Case Upcase(Ch) of
    'T': TFScreen(Sys1,Flag1);
    'B': BTScreen(Work,Flag2,Chol);
    'Q': Exit;                           {To Main Program}
  end;
End;                                     {Procedure Iterate}

{===========================Main Program==============================}
Begin
{===========================Initialization===========================}
  Rstep1:=2*Sz;
  Cstep1:=Power(Sz,2) + 1;
  Rstep2:=2*Power(Sz,3);
  Cstep2:=((Power(Sz,4)+Power(Sz,2))div 2)+Power(Sz,2)+1;
  For I:= 0 to Nmod do begin
    Cbar[I]:= 0;
    Padic[I,1]:=0;
    Padic[I,2]:=0;
  end;
  For I:= 1 to CStep2 do
    Chol[I]:=0;
  For I:= 1 to 6 do begin
    For J:= 1 to 6 do
      Trace[I,J]:= 0;
  end;
  For I:= 1 to Rstep1 do begin
    For J:= 1 to Cstep1 do begin
      Sys1[I,J]:= 0;
      Sys2[I,J]:= 0;
    end;
```

```
  end;                                        {Initialization}
  Flag1:=0;
  Flag2:=0;
  For I:= 1 to Rstep2 do begin
    For J:= 1 to Cstep2 do begin
      Raw[I,J]:= 0;
      Work[I,J]:= 0;
    end;
  end;
  For I:= 1 to Sz do begin
    For J:= 1 to Sz do begin
      For K:= 1 to Sz do begin
        M[I,J,K]:= 0;
        N[I,J,K]:= 0;
        For U:= 1 to Sz do begin
          Rt1[I,J,K,U]:=0;
          Rt2[I,J,K,U]:=0;
          For V:= 1 to Sz do begin
            Lft1[I,J,K,U,V]:= 0;
            Lft2[I,J,K,U,V]:= 0;
          end;
        end;
      end;
    end;
  end;
{============================Execution================================}
  Inp(M,N);
  Invint(Cbar);
  Invval(Padic);
  Tracer(M,N,Trace);
  Sysbuilder(Sys1,Sys2,Trace);
  LeftSide(Lft1,Lft2,M);
  RightSide(Rt1,Rt2,N);
  ColTracker(Chol);
  BigTableau(Raw,Work,Rt1,Rt2,Lft1,Lft2,Chol);
  Write(Lst,'Trace Formula Matrices');
  GotoXY(40,WhereY);
  Writeln(Lst,'Sys2 For Information Only');
  For I:= 1 to Sz do begin
    For J:= 1 to Sz do
      Write(Lst,'  ',J,I);
  end;
  Write(Lst,' Col');
  Writeln(Lst);Writeln(Lst);
  For I:=1 to Rstep1 do begin               {Echo of Sys1 and Sys2}
    For J:= 1 to Cstep1 do
      Write(Lst,Sys1[I,J]:4);
     GotoXY(40,WhereY);
    Write(Lst,'               ');
    For J:=1 to Cstep1 do
      Write(Lst,Sys2[I,J]:4);
    Writeln(Lst);
```

```
end;                                   {Echo of Sys1 and Sys2}
Reduce(Sys2,Cbar,rstep1,cstep1,Padic);
REP:Writeln(Lst);
Reduce(Sys1,Cbar,rstep1,cstep1,Padic);
Writeln(Lst,'Reduced Trace Formula Matrices');
For I:=1 to Rstep1 do begin
  For J:= 1 to Cstep1 do
    Write(Lst,Sys1[I,J]:4);
  Write(Lst,'              ');
  For J:=1 to Cstep1 do
    Write(Lst,Sys2[I,J]:4);
  Writeln(Lst);
end;
Flag1:=0;
Iterate(Sys1,Work,Flag1,Flag2,Chol);
If Flag1=1 then
  Goto REP;
Runnum:=1;
REP2:If Runnum = 1 then begin
  Writeln(Lst);
  Writeln(Lst,'Raw Tableau entries.');
  For I:= 1 to Rstep2 do begin
    For J3:= 1 to Cstep2 do begin
      If Work[I,J3]<>0 then begin
        For J:= 1 to Cstep2 do
          Write(Lst,Work[I,J]:2);
        Writeln(Lst);
        Goto LOOPI;
      end;
    end;
  LOOPI:end;
end;
Writeln(Lst);
Reduce(Work,Cbar,Rstep2,Cstep2,Padic);
Writeln(Lst,'Run Number ',Runnum);
Writeln(Lst,'Reduced Tableau entries.');
For I:= 1 to Rstep2 do begin
    For J3:= 1 to Cstep2 do begin
      If Work[I,J3]<>0 then begin
        For J:= 1 to Cstep2 do
          Write(Lst,Work[I,J]:2);
        Writeln(Lst);
        Goto LOOPI2;
      end;
    end;
  LOOPI2:end;
Flag2:=0;
Iterate(Sys1,Work,Flag1,Flag2,Chol);
If Flag2=1 then begin
  Runnum:= Runnum+1;
  Goto REP2;
end;
```

```
End.                                     {Program}
```

```
Program IDEMPOT;
{This Program is a subset of QUADID.  Its purpose is to simplify the
search for an idempotent other than 1 in a ring R.
{==========================Declarations============================}
Label
 REP,REP2,LOOPI,LOOPI2;
Const
  Sz = 3;                             {Rank R}
  Nmod = 3;                           {Char R}
  Base = 3;
Type
  Blk = array[0..Nmod] of Integer;
  Sys = array[1..18,1..34] of Integer;
  Cube = array[1..Sz,1..Sz,1..Sz] of Integer;
  Vec = array[0..Nmod,1..2]of Integer;
  LftSd = Array[1..Sz,1..Sz,1..Sz,1..Sz,1..Sz] of Integer;
  Rtsd = Array[1..Sz,1..Sz,1..Sz,1..Sz] of Integer;
  Stak = Array[1..24] of Integer;
  Tr = Array[1..10,1..6] of Integer;
Var
  I,J,J3,K,U,V,rstep1,rstep2,cstep1,cstep2,Flag1,
  Flag2,Rank,Runnum: Integer;
  Sys1,Raw,Work: Sys;
  M,N: Cube;
  Cbar: Blk;
  Padic: Vec;
  Lft1,Lft2: LftSd;
  Rt1,Rt2: Rtsd;
  Chol: Stak;                         {# of Quadratic Variables}
  Trace: Tr;
  Cent: Char;
{===========================Function Power==========================}
Function Power( B,v:Integer):Integer;
Begin
  Power:=Round(Exp(V*Ln(B)));
end;                                  {Function Power}
{===========================Procedure Inp===========================}
Procedure Inp (var M:Cube);
  Var I,J,K,Num: Integer;
begin
  ClrScr;
  GotoXY(8,4);
  Writeln('The characteristic of R is ',Nmod,'; its rank is ',Sz,'.');
  Writeln('To change either, edit the program constants Nmod or Sz.');
  GotoXY(8,8);
  Writeln('Enter [M] by rows, assuming [M] has been checked by
           BASPROPS.');
  GotoXY(1,10);
  For I:= 1 to Sz do begin
    For K:= 1 to Sz do begin
      For J:= 1 to Sz do begin
        Read(Kbd,Num);
```

```
        M[I,J,K]:= Num;
        GotoXY((3*J),WhereY);
        Write(Num);
      end;
      Writeln;
    end;
  end;
  GotoXY(1,20);
  Write(' Input of Cube Complete.');
  Delay(1000);
  ClrScr;
end;                                   {Procedure Inp}
{===========================Procedure Searchtype=====================}
Procedure Searchtype(var Cent:Char; var Rank:Integer);
Label
  ONE,TWO;
Begin
  ONE:GotoXY(5,8);
  Writeln('Find a [C]entral or [N]oncentral Idempotent?');
  Read(Kbd,Cent);
  If Not (Upcase(Cent) in ['C','N']) then
    Goto ONE;
  GotoXY(5,9);
  Write(Cent);
  TWO:GotoXY(5,12);
  Writeln('Rank of eR is [1],[2],..,or [k]?');
  Read(Kbd,Rank);
  If Not (Rank in [1..Sz]) then
    Goto TWO;
  GotoXY(5,14);
  Write(Rank);
end;                                   {Procedure Searchtype}
{===========================Procedure Tracer========================}
Procedure Tracer(M,N:Cube; var Trace:Tr; Cent:Char; Rank: Integer);
Var
  Sum1 ,Sum_1,Sum2_,Sum_2,I,J: Integer;
begin
  For I:= 1 to Sz do begin
    Sum1_:= 0;
    Sum_1:= 0;
    Sum2_:= 0;
    Sum_2:= 0;
    For J:= 1 to Sz do begin
      Sum1_:=Sum1_ + M[I,J,J];
      Sum_1:=Sum_1 + M[J,I,J];
      If I = 1 then begin
        Sum2_:=Sum2_ + N[1,J,J];
        Sum_2:=Sum_2 + N[J,1,J];
      end;
    end;
    Trace[I,5]:=Sum1_ mod Nmod;
    Trace[5,I]:=Sum_1 mod Nmod;
```

```
    If I = 1 then begin
      Trace[1,6]:=Sum2_ mod Nmod;
      Trace[6,1]:=Sum_2 mod Nmod;
    end;
  end;
end;                                    {Procedure Trace}
{===========================Procedure SysBuilder=====================}
Procedure SysBuilder(var Sys1:sys; Trace:Tr; Cent:Char);
Var
  J: Integer;
begin
  For J:= 1 to Sz do begin
    Sys1[1,J]:= Trace[J,5];
    Sys1[2,J]:= Trace[5,J];
  end;
  Sys1[1,(Sz*Sz + 1)]:= Trace[1,6];
  Sys1[2,(Sz*Sz + 1)]:= Trace[6,1];
end;                                    {Procedure SysBuilder}
{===========================Procedure Invint========================}
Procedure Invint (var cbar: Blk);
Var
  I,J: Integer;
Label
  ONE;
Begin
  Cbar[1]:= 1;
  For I:= 2 to (Nmod - 1) do begin
    If Cbar[I] = 0 then begin
      For J:= 2 to (Nmod - 1) do begin
        If (I*J) mod Nmod = 1 then begin
          Cbar[I]:= J;
          Cbar[J]:= I;
          Goto ONE;
        end;
      end;
    end;
  ONE:end;
end;                                    {Procedure Invint}
{===========================Procedure Invval========================}
{If X = (P**K)*U, then Padic[X,1] = K, Padic[X,2] = U.  If X = 0, then
Padic[X,2] = Expo.}
Procedure Invval(var Padic:Vec);
Var
  K,I,Expo: Integer;
Begin
  Expo:= Round(Ln(Nmod)/Ln(Base));
  Padic[0,1]:=Expo;                     {Padic[0,2] is already 0}
  For I:= 1 to (Nmod-1) do begin
    K:= 1;
    While (I mod Power(Base,K) = 0) and (K <= Expo) do
      K:= K+1;
    Padic[I,1]:= K-1;
```

```
    Padic[I,2]:= I div Power(Base,(K-1));
  end;
end;                                   {Procedure Invval}
{===========================Procedure LeftSide======================}
Procedure LeftSide(var Lft1,Lft2: LftSd; M: Cube; Cent:Char);
Var
  L,C,U,V: Integer;
Begin
  For L:= 1 to Sz do begin
    For V:= 1 to Sz do begin
      For U:= 1 to Sz do begin
        For C:= 1 to Sz do begin
          Lft1[L,V,1,U,C]:= M[V,U,L];
          If Upcase(Cent) = 'C' then
            Lft2[L,V,1,U,C]:= M[U,V,L];
        end;
      end;
    end;
  end;
end;                                   {LeftSide}
{===========================Procedure RightSide=====================}
Procedure RightSide(var Rt1,Rt2:Rtsd; N: Cube; Cent:Char);
Var
  L,C,U: Integer;
Begin
  For L:= 1 to Sz do begin
    For C:= 1 to Sz do begin
      For U:= 1 to Sz do begin
        Rt1[1,L,C,U]:= N[1,C,U];
        If Upcase(Cent) = 'C' then
          Rt2[1,L,C,U]:= N[C,1,U];
      end;
    end;
  end;
end;                                   {Procedure RightSide}
{===========================Procedure Coltracker====================}
{Arranges quadratic variables in lexicographic order}
Procedure ColTracker (var Chol: Stak);
Var
  Knt,U,C,V: Integer;
Begin
  Knt:=0;
  For V:= 1 to Sz do begin
    For U:= 1 to Sz do begin
      For C:= 1 to Sz do begin
        If (10*V+1)<=(10*U+C) then begin
          Knt:= Knt + 1;
          Chol[Knt]:= C + 10*(U + 10*(1 + 10*V));  {V1UC as Base 10 #}
        end;
      end;
    end;
  end;
```

```
End;                                      {Procedure Coltracker}
{==========================Procedure BigTableau=====================}
Procedure BigTableau (var Raw,Work:Sys; Rt1,Rt2:Rtsd;
                      Lft1,Lft2:LftSd; Chol:Stak; Cent:Char);
Var
  C,L,U,V,Check1,Check2,Ro,Nro,Kol,QuadCols,J,LastCol: Integer;
Label
  ONE,TWO;
Begin
  QuadCols:=(2*Power(Sz,3)-Power(Sz,2)+Sz)div 2;
  LastCol:= QuadCols + Power(Sz,2) + 1;
  For L:= 1 to Sz do begin                {Entries from LeftSide}
   For V:= 1 to Sz do begin
     For U:= 1 to Sz do begin
      For C:= 1 to Sz do begin
       Ro:= C+Sz*(L-1);
       Check1:=C+10*(U+10*(1+10*V));
       Check2:=1+10*(V+10*(C+10*U));
       For J:= 1 to QuadCols do begin
        If(Chol[J]=Check1) or (Chol[J]=Check2) then begin
         Raw[Ro,J]:=(Raw[Ro,J]+Lft1[L,V,1,U,C])mod Nmod;
         Work[Ro,J]:= Raw[Ro,J];
         If Upcase(Cent) = 'C' then begin
          Nro:= Ro + Sz*Sz;
          Raw[Nro,J]:= (Raw[Nro,J]+Lft2[L,V,1,U,C])mod Nmod;
          Work[Nro,J]:= Raw[Nro,J];
         end;
         Goto TWO;
        end;
       end;                               {j}
      TWO:end;                            {c}
     end;                                 {u}
   end;                                   {v}
  end;                                    {Entries from LeftSide}
  For L:= 1 to Sz do begin                {Entries from RightSide}
    For C:= 1 to Sz do begin
     For U:= 1 to Sz do begin
      Ro:= C+Sz*(L-1);
      Kol:= QuadCols+U+(L-1)*Sz;
      Raw[Ro,Kol]:= (Nmod-Rt1[1,L,C,U])mod Nmod;
      Work[Ro,Kol]:= Raw[Ro,Kol];
      If Upcase(Cent) = 'C' then begin
       Nro:= Ro + Sz*Sz;
       Raw[Nro,Kol]:= (Nmod-Rt2[1,L,C,U])mod Nmod;
       Work[Nro,Kol]:= Raw[Nro,Kol];
     end;
    end;
   end;
  end;                                    {Entries from RightSide}
End;                                      {Procedure BigTableau}
{==========================Procedure Reduce========================}
Procedure Reduce (var A:sys; cbar:Blk; Bot,LastCol:Integer; Padic:vec);
```

```
Label
  ONE,TWO,THR;
Var
  I,I3,J,J3,Piv,J1,L1,L2,Diff,Mult,Temp,Minpower,Minrow,Expo: Integer;
Begin
 Expo:= Round(Ln(Nmod)/Ln(Base));
 J:=0;
 ClrScr;
 For I:= 1 to (Bot -1)do begin          {Loop}
  TWO:If J < LastCol then
   J:=J+1
  else
   Goto THR;
  Piv:= I;
  Minpower:= Expo;
  Minrow:= I;
  While(Padic[A[Piv,J],1] >= 1) and (piv <= Bot) and (J<= LastCol)do
  begin                            {Pivot Search}
   If (Piv = I) then
    Minpower:= Padic[A[Piv,J],1];
   If Piv > I then begin
    If Padic[A[Piv,J],1] < Padic[A[(Piv-1),J],1] then begin
     Minpower:= Padic[A[Piv,J],1];
     Minrow:= Piv;
    end;
   end;
   Piv:= Piv + 1;
  end;                              {Pivot Search}
  If (Piv > Bot) and (Minpower=Expo) then
   Goto TWO;
  If (Piv > Bot) and (Minpower < Expo) then
   Piv:= Minrow;
  If I < Piv then begin             {Pivot}
   For J1:= 1 to LastCol do begin
    Temp:= A[I,J1];
    A[I,J1]:=A[Piv,J1];
    A[Piv,J1]:= Temp;
   end;
  end;                              {Pivot}
  For L1:= (I + 1) to Bot do begin    {Reduce Bot L1}
   Diff:= Power(Base,(Abs(Padic[A[L1,J],1]-Padic[A[I,J],1])));
   Mult:=((Diff*Cbar[Padic[A[I,J],2]]mod Nmod)
         *Padic[A[L1,J],2])mod Nmod;
   For L2:= J to LastCol do
    A[L1,L2]:=((A[L1,L2] - ((Mult*A[I,L2]) mod Nmod))+ Nmod) mod Nmod;
  end;                              {Reduce Bot L1}
 end;                                {Loop}
 THR:For I3:= 1 to Bot do begin
  For J3:= 1 to LastCol do begin
   If A[I3,J3]<>0 then begin
    Mult:= Cbar[Padic[A[I3,J3],2]];
    For J1:=J3 to LastCol do
```

```
     A[I3,J1]:= (Mult*A[I3,J1])mod Nmod;
    Goto ONE;
   end;
  end;
 ONE:end;
end;                                    {Procedure Reduce}
{==========================Procedure BTDep==========================}
Procedure BTDep(var A:Sys; Chol:Stak);
Label
  LOOPC;
Var
 I2,J2,D1,D2,QuadCols,Bot,LastCol,Kol,Dest,Sig,Tau,
 Num,J,K,U,C,Ro,Check1,Check2: Integer;
Begin
 Bot:= 2*Power(Sz,2);
 QuadCols:= (2*Power(Sz,3)-Power(Sz,2)+Sz)div 2;
 LastCol:= Quadcols + Power(Sz,2) + 1;
 ClrScr; GotoXY(1,8);
 Writeln('Modding Big Tableau. Assume you know A(Sig,Tau) =
                              D1-D2*A(I2,J2).');
 Write ('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
 Write ('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
 Write ('I2 = ');   Read(Num);   I2:= Num;   Writeln;
 Write ('J2 = ');   Read(Num);   J2:= Num;   Writeln;
 Write ('D1 = ');   Read(Num);   D1:= Num;   Writeln;
 Write ('D2 = ');   Read(Num);   D2:= Num;   Writeln;
 Writeln('Working.');
 Kol:= QuadCols + Tau + Sz*(Sig-1);
 For U:= 1 to Sz do begin
  For C:= 1 to Sz do begin
   For J:= 1 to QuadCols do begin
    Check1:=C+10*(U+10*(Tau+10*Sig));
    Check2:=Tau+10*(Sig+10*(C+10*U));
    If(Chol[J]=Check1) or (Chol[J]=Check2) then begin
     Dest:= QuadCols + C + Sz*(U-1);
     For Ro:= 1 to Bot do
      A[Ro,Dest]:=(A[Ro,Dest]+D1*A[Ro,J])mod Nmod;
     For K:= 1 to QuadCols do begin
      Check1:=C+10*(U+10*(J2+10*I2));
      Check2:=J2+10*(I2+10*(C+10*U));
      If(Chol[K]=Check1) or (Chol[K]=Check2) then begin
       For Ro:= 1 to Bot do begin
        A[Ro,K]:=((A[Ro,K]-D2*A[Ro,J])mod Nmod + Nmod)mod Nmod;
        A[Ro,J]:=0;
       end;
       Goto LOOPC;
      end;          {k}
     end;           {Ro}
    end;            {Check#}
   end;             {j}
  LOOPC:end;
 end;
```

```
End;                                    {Procedure BTDep}
{=========================Procedure BTCon=========================}
Procedure BTCon(var A:Sys; Chol:Stak);
Label
 LOOPC;
Var
 QuadCols,Bot,LastCol,Kol,Dest,Sig,Tau,
 Num,Con,J,U,C,Ro,Check1,Check2: Integer;
Begin
 Bot:= 2*Power(Sz,2);
 QuadCols:= (2*Power(Sz,3)-Power(Sz,2)+2)div 2;
 LastCol:= Quadcols + Power(Sz,2) + 1;
 ClrScr; GotoXY(1,8);
 Writeln('Modding Big Tableau.  Assume you know A(Sig,Tau) = Con.');
 Write ('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
 Write ('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
 Write ('Con = ');   Read(Num);   Con:= Num;   Writeln;
 Writeln('Working.');
 Kol:= QuadCols + Tau + Sz*(Sig-1);
 For U:= 1 to Sz do begin
  For C:= 1 to Sz do begin
   For J:= 1 to QuadCols do begin
    Check1:= C+10*(U+10*(Tau+10*Sig));
    Check2:= Tau+10*(Sig+10*(C+10*U));
    If(Chol[J]=Check1)or(Chol[J]=Check2) then begin
     Dest:= QuadCols + C + Sz*(U-1);
     For Ro:= 1 to Bot do begin
      A[Ro,Dest]:=(A[Ro,Dest]+Con*A[Ro,J])mod Nmod;
      A[Ro,J]:=0;
     end;
     Goto LOOPC;
    end;
   end;
  LOOPC:end;
 end;
 For Ro:= 1 to Bot do begin
  A[Ro,LastCol]:=((Nmod-Con*A[Ro,Kol])mod Nmod + Nmod)mod Nmod;
  A[Ro,Kol]:=0;
 end;
End;                                    {Procedure BTCon}
{=========================Procedure BTScreen======================}
Procedure BTScreen(var Work:Sys; var Flag2: Integer; Chol:Stak);
Label
 REP;
Var
  Ch:Char;
Begin
  REP:ClrScr;
  GotoXY(1,8);
  Writeln('Choose one of the following options:');
  Writeln('Change Variable to [C]onstant.');
  Writeln('Re-express [D]ependent Variable in terms of one other.');
```

```
    Writeln('[R]educe Big Tableau again.');
    Writeln('[Q]uit Modifying Big Tableau.');
    Repeat
      Read(Kbd,Ch)
    Until Upcase(Ch) in ['C','D','R','Q'];
    Case Upcase(Ch) of
      'C': begin
             BTCon(Work,Chol);
             Goto REP;
           end;
      'R': Flag2:= 1;
      'D': begin
             BTDep(Work,Chol);
             Goto REP;
           end;
      'Q': Exit;                         {To Iterate}
    end;
  End;                                   {Procedure BTScreen}
  {============================Procedure TF=============================}
  Procedure TF(var Sys1:Sys);
  Var
    Col,Num,K,Sig,Tau,Con,I: Integer;
  Begin
    Col:= Power(Sz,2)+1;
    ClrScr;
    GotoXY(1,8);
    Writeln('Modifying Trace Formula. Assumes you know A(Sig,Tau) =
              Con.');
    Write('Sig = ');   Read(Num);   Sig:= Num;   Writeln;
    Write('Tau = ');   Read(Num);   Tau:= Num;   Writeln;
    Write('Con = ');   Read(Num);   Con:= Num;   Writeln;
    Writeln('Working.');
    K:=Sz*(Sig-1) + Tau;
    For I:= 1 to 2 do begin
      Sys1[I,Col]:= ((Sys1[I,Col]-Con*Sys1[I,K])mod Nmod + Nmod)mod Nmod;
      Sys1[I,K]:=0;
    end;
  End;                                   {Procedure TF}
  {============================Procedure TFScreen======================}
  Procedure TFScreen(var Sys1:Sys; var Flag1: Integer);
  Label
    ONE;
  Var
    Ch: Char;
  Begin
    ONE:ClrScr;
    GotoXY(8,8);
    Writeln('Choose one of the following options:');
    Writeln('Change Variable to [C]onstant.');
    Writeln('[R]educe Trace Formula Again.');
    Writeln('[Q]uit.');
    GotoXY(8,10);
```

```
  Repeat
    Read(Kbd,Ch);
  Until Upcase(Ch) in ['C','R','Q'];
  Case Upcase(Ch) of
    'C': begin
           TF(Sys1);
           Goto ONE;
         end;
    'Q': Exit;                               {To Iterate}
    'R': Flag1:=1;
  end;
End;                                         {Procedure TFScreen}

{============================Procedure Iterate========================}
Procedure Iterate(var Sys1,Work:Sys;var Flag1,Flag2:Integer;Chol:Stak);
Var
  Ch: Char;
Begin
  ClrScr;
  GotoXY(10,8);
  Writeln('Modify [T]race Formulas, [B]ig Tableau, or
                [Q]uit Modifying.');
  GotoXY(10,10);
  Repeat
    Read(Kbd,Ch)
  Until Upcase(Ch) in ['T','B','Q'];
  Case Upcase(Ch) of
    'T': TFScreen(Sys1,Flag1);
    'B': BTScreen(Work,Flag2,Chol);
    'Q': Exit;                               {To Main Program}
  end;
End;                                         {Procedure Iterate}

{============================Main Program=============================}
Begin
{============================Initialization===========================}
  Rstep1:=2;
  Cstep1:=Power(Sz,2) + 1;
  Rstep2:=2*Power(Sz,2);
  Cstep2:=((2*Power(Sz,3)-Power(Sz,2)+Sz)div 2)+Power(Sz,2)+1;
  For I:= 0 to Nmod do begin
    Cbar[I]:= 0;
    Padic[I,1]:=0;
    Padic[I,2]:=0;
  end;
  For I:= 1 to CStep2 do
    Chol[I]:=0;
  For I:= 1 to 6 do begin
    For J:= 1 to 6 do
      Trace[I,J]:= 0;
  end;
  For I:= 1 to Rstep1 do begin
```

```
    For J:= 1 to Cstep1 do begin
      Sys1[I,J]:= 0;
    end;
  end;                                {Initialization}
  Flag1:=0;
  Flag2:=0;
  For I:= 1 to Rstep2 do begin
    For J:= 1 to Cstep2 do begin
      Raw[I,j]:= 0;
      Work[I,J]:= 0;
    end;
  end;
  For I:= 1 to Sz do begin
    For J:= 1 to Sz do begin
      For K:= 1 to Sz do begin
        M[I,J,K]:= 0;
        N[I,J,K]:= 0;
        For U:= 1 to Sz do begin
          Rt1[I,J,K,U]:=0;
          Rt2[I,J,K,U]:=0;
          For V:= 1 to Sz do begin
            Lft1[I,J,K,U,V]:= 0;
            Lft2[I,J,K,U,V]:= 0;
          end;
        end;
      end;
    end;
  end;
{============================Execution=================================}
  Inp(M);
  Invint(Cbar);
  Invval(Padic);
  Searchtype(Cent,Rank);
  For I:= 1 to Rank do begin
    N[1,I,I]:= 1;
    N[I,1,I]:= 1;
  end;
  Tracer(M,N,Trace,Cent,Rank);
  Sysbuilder(Sys1,Trace,Cent);
  LeftSide(Lft1,Lft2,M,Cent);
  RightSide(Rt1,Rt2,N,Cent);
  ColTracker(Chol);
  BigTableau(Raw,Work,Rt1,Rt2,Lft1,Lft2,Chol,Cent);
  Writeln(Lst,'Trace Formula Matrix');
  Writeln(Lst);
  For I:= 1 to Sz do begin
    For J:= 1 to Sz do
      Write(Lst,'  ',J,I);
  end;
  Write(Lst,' Col');
  Writeln(Lst);Writeln(Lst);
  For I:=1 to Rstep1 do begin         {Echo of Sys1}
```

```
    For J:= 1 to Cstep1 do
      Write(Lst,Sys1[I,J]:4);
    Writeln(Lst);
  end;                                   {Echo of Sys1}
  REP:Writeln(Lst);
  Reduce(Sys1,Cbar,rstep1,cstep1,Padic);
  Writeln(Lst,'Reduced Trace Formula Matrix');
  For I:=1 to Rstep1 do begin
    For J:= 1 to Cstep1 do
      Write(Lst,Sys1[I,J]:4);
    Writeln(Lst);
  end;
  Flag1:=0;
  Iterate(Sys1,Work,Flag1,Flag2,Chol);
  If Flag1=1 then
    Goto REP;
  Runnum:=1;
  REP2:If Runnum = 1 then begin
    Writeln(Lst);
    Writeln(Lst,'Raw Tableau entries.');
    For I:= 1 to Rstep2 do begin
      For J3:= 1 to Cstep2 do begin
        If Work[I,J3]<>0 then begin
          For J:= 1 to Cstep2 do
            Write(Lst,Work[I,J]:2);
          Writeln(Lst);
          Goto LOOPI;
        end;
      end;
    LOOPI:end;
  end;
  Writeln(Lst);
  Reduce(Work,Cbar,Rstep2,Cstep2,Padic);
  Writeln(Lst,'Run Number ',Runnum);
  Writeln(Lst,'Reduced Tableau entries.');
  For I:= 1 to Rstep2 do begin
      For J3:= 1 to Cstep2 do begin
        If Work[I,J3]<>0 then begin
          For J:= 1 to Cstep2 do
            Write(Lst,Work[I,J]:2);
          Writeln(Lst);
          Goto LOOPI2;
        end;
      end;
    LOOPI2:end;
  Flag2:=0;
  Iterate(Sys1,Work,Flag1,Flag2,Chol);
  If Flag2=1 then begin
    Runnum:= Runnum+1;
    Goto REP2;
  end;
End.                                     {Program}
```

END
12-87
DTIC